BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-036529
(43)Date of publication of application : 07.02.1995

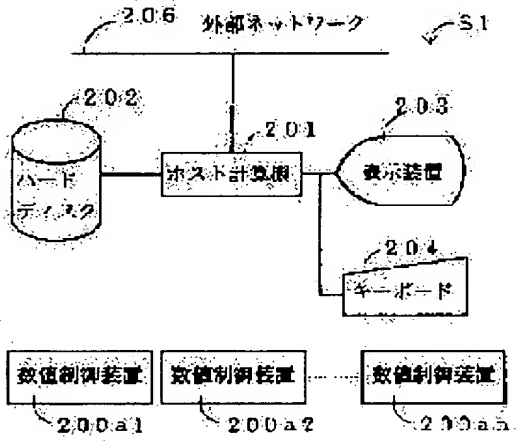
(51)Int.Cl. G05B 19/4155
G06F 9/46

(21)Application number : 05-175490
(22)Date of filing : 15.07.1993

(71)Applicant : MITSUBISHI ELECTRIC CORP
(72)Inventor : NAMIKADO SHIGEKI

(54) EXECUTION SYSTEM FOR CONTROL SOFTWARE OF NUMERICAL CONTROLLER

(57)Abstract:
PURPOSE: To effectively carry out the control software at a low cost by allocating the task functions of a numerical controller to the threads in a process or the tasks which are carried out in an operating system of the general-purpose multitask or multithread.
CONSTITUTION: A control software execution system S1 consists of the numerical controllers 200a1-200an, a host computer 201, a hard disk 202, a display device 203, and a keyboard 204. The controller 200a1-200an carry out the control software in a real-time operating system and attain the functions of the numerical controllers. The computer 201 carries out the control software of the numerical controllers in a general-purpose multithread operating system. The disk 202 functions as an external storage of the computer 201, and the device 203 shows various necessary information for users.



LEGAL STATUS

[Date of request for examination] 05.12.1996
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number] 2820189
[Date of registration] 28.08.1998
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（JP）　(12)特　許　公　報（B2）　(11)特許番号

第2820189号

(45)発行日　平成10年(1998)11月5日　(24)登録日　平成10年(1998)8月28日

(51)Int.Cl.[6]	識別記号	FI	
G05B 19/4155		G05B 19/403	X
G06F 9/46	340	G06F 9/46	340B

請求項の数6（全 30 頁）

(21)出願番号　特願平5－175490

(22)出願日　平成5年(1993)7月15日

(65)公開番号　特開平7－36529

(43)公開日　平成7年(1995)2月7日

審査請求日　平成8年(1996)12月5日

(73)特許権者　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目2番3号

(72)発明者　南角　茂樹
名古屋市東区矢田南五丁目1番14号　三菱電機株式会社　名古屋製作所内

(74)代理人　弁理士　宮田　金雄　（外2名）

審査官　千葉　成就

(58)調査した分野(Int.Cl.[6]，DB名)
G05B 19/4155
G06F 9/46 340

(54)【発明の名称】　数値制御装置の制御ソフトウェア実行システム

(57)【特許請求の範囲】

【請求項1】　リアルタイムオペレーティングシステム上で制御ソフトウェアを実行したときに起動される複数のタスクの共同作業によって数値制御装置としての機能を実現する数値制御装置のタスクの機能を、汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステム上で実行するプロセスまたはタスク中のスレッドに割り当てると共に、スレッド間のスケジューリングの制御用に1つのスレッドを割り当てることを特徴とする数値制御装置の制御ソフトウェア実行システム。

【請求項2】　請求項1の数値制御装置の制御ソフトウェア実行システムにおいて、
汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステムを用いた計算機と、その計算機に通信回線を介して接続された数値制御装置とを具備してなり、前記計算機と前記数値制御装置の間に分散型共有メモリの機能を持たせ、前記数値制御装置のタスクの機能を、前記分散型共有メモリの機能を利用して、前記計算機と前記数値制御装置とで分散して処理することを特徴とする数値制御装置の制御ソフトウェア実行システム。

【請求項3】　請求項1または請求項2の数値制御装置の制御ソフトウェア実行システムにおいて、汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステムを用いた計算機と、その計算機に通信回線を介して接続された数値制御装置とを具備してなり、前記計算機は前記数値制御装置への割り込み要求の有無を監視することを特徴とする数値制御装置の制御ソフトウェア実行システム。

【請求項４】　請求項１から請求項３のいずれかの数値制御装置の制御ソフトウェア実行システムにおいて、
複数のＣＰＵカードを有するＣＰＵプールを具備し、そのＣＰＵカード中のＣＰＵを前記数値制御装置のＣＰＵとして割り当てることを特徴とする数値制御装置の制御ソフトウェア実行システム。
【請求項５】　請求項４の数値制御装置の制御ソフトウェア実行システムにおいて、
前記ＣＰＵカード中のＣＰＵを前記複数の数値制御装置のＣＰＵとして割り当てることを特徴とする数値制御装置の制御ソフトウェア実行システム。
【請求項６】　請求項５の数値制御装置の制御ソフトウェア実行システムにおいて、
前記ＣＰＵプール中の１つのＣＰＵカードを割り込み検査専用のＣＰＵカードとして使用することを特徴とする数値制御装置の制御ソフトウェア実行システム。

【発明の詳細な説明】
【０００１】
【産業上の利用分野】この発明は、数値制御装置の制御ソフトウェア実行システムに関し、さらに詳しくは、数値制御装置の制御ソフトウェア（コントロールプログラム）を汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステム上で実行できる数値制御装置の制御ソフトウェア実行システムに関する。
【０００２】
【従来の技術】図２７に示す数値制御装置５０１のハードウェア構成図を用いて従来の数値制御装置の一般的構成を説明する。１は、各部の作動を制御したり、数値制御に必要な演算を行なったりするメインＣＰＵである。２は、各部を結ぶシステムバスである。３は、数値制御装置の主要機能を実現する制御ソフトウェアなどを格納するＲＯＭ（不揮発性の記憶装置）である。４は、一時記憶やワークエリアなどに用いるＲＡＭ（揮発性の記憶装置）である。５は、外部との間でシリアル通信によりデータのやり取りを行なうＳＩＯインタフェース部である。６は、数値制御装置の運転状態を表示したり、数値制御装置に与えた指令を確認したりするための表示装置である。７は、数値制御装置に指令を与えるためのキーボード（入力装置）である。８は、サーボモータを制御するための指令を演算するサーボ制御部である。このサーボ制御部８は、メインＣＰＵ１とは別の専用ＣＰＵを備えてもよい。９は、サーボ制御部８から受け取った指令を増幅してサーボモータへ駆動信号を出力するサーボアンプである。１０は、工作機械（図示せず）の加工部を制御するためのサーボモータである。１１は、工作機械との間で、サーボ制御指令以外のデータをやり取りするためのプログラムコントローラ部である。１２は、メインＣＰＵ１に与えられる、システムクロック（図示せず）と外部割り込み信号を表している。システムクロックは、数値制御装置全体を制御するためのクロック信号である。また、外部割り込み信号は、電源異常や非常停止などのイベント（緊急な出来事）の発生をメインＣＰＵ１に通知するための信号である。
【０００３】次に、上記数値制御装置５０１の動作を説明する。まず、メインＣＰＵ１は、ＲＯＭ３に書き込まれている制御ソフトウェアをシステムバス２を通して１命令ずつ順に読み込んで実行する。図２８に、命令読込後の処理手順を示す。加工プログラム入力処理２１では、ＳＩＯインタフェース部５を通して外部から加工プログラム２０を読み込み、ＲＡＭ４に格納する。そして、加工プログラム２０をブロック（所定の単位）ごとに内部データに変換する。補正計算処理２２では、ブロックごとの内部データを処理し、増分移動量を算出する。また、工具径や工具長などを補正する。さらに、内部の座標値の更新処理を行なう。設定表示処理２３では、数値制御装置の各種データを表示装置６に表示する。また、キーボード７を用いてオペレータが入力した各種設定データをＲＡＭ４に格納する。補間処理２４では、補正計算処理２２の処理結果を用いて、微小時間ごとの各軸の移動量を算出する。サーボ処理２５では、補間処理２４の処理結果を用いて、さらに小さな単位時間ごとの各軸の移動量に変換する。更に、サーボモータ１０からのフィードバック制御（図示せず）を行なう。プログラムコントロール処理２６では、工作機械との間での入出力処理や主軸の制御など、工作機械の周辺の制御などを行なう。
【０００４】さて、上記の数値制御装置５０１は、図２７を用いて先に説明したようにメインＣＰＵ１に外部割り込み信号１２を入力して割込み処理を行なわせることで、非常停止などの緊急事態に対処することが出来る。メインＣＰＵ１は、外部割り込み信号１２（図２７）が入力されると、予め指定された別処理を実行し、それら別処理が終了した後に、通常の命令に復帰する。
【０００５】図２９は、割り込み処理の概念図である。３０～３６は、通常の命令である。３７～４０は、割り込み命令である。４１は、通常の命令への復帰命令である。例えば、メインＣＰＵ１が通常の命令３３を実行中に外部割り込み信号１２（図２７）が入力されると、メインＣＰＵ１は、通常の命令３３の処理終了後、割り込みを検出し、予め指定されていた割り込み命令３７の実行を開始する。そして、割り込み命令３７～４０の実行を終了した後、復帰命令４１を実行して、通常の命令３４に復帰し、通常の命令３５，３６を実行する。なお、割り込み処理の処理時間が長くなると多重割り込みなどを考慮する必要が出てくるので、割り込み処理の処理時間をできるだけ短くすることが望ましい。
【０００６】ところで、上記の数値制御装置５０１の制数値制御装置の制御ソフトウェアにより実現すべき機

能の多様性のためソフトウェア量も大きくなり、多人数数値制御装置の制御ソフトウェアの機能ごとに、処理に必要な応答時間（ターンアラウンドタイム，デッドライン）が異なる。例えば、サーボ処理２５（図２８）は、処理結果の算出が遅れると切削が止って被加工物が不良品になってしまうので、実時間処理でなくてはならない。一方、表示装置６への表示処理などは、多少遅れても不都合は生じないので、実時間処理でなくてもよメインＣＰＵ１が実行すべき割り込み処理に、多くの種類がある。

これらの理由により、数値制御装置５０１の制御ソフトウェアは、一般にリアルタイムオペレーティングシステムの制御のもとで実行される機能ごとのタスクを実行単位とすることが多い。

【０００７】次に、リアルタイムオペレーティングシステムにより各タスクを制御する方法を説明する。あるタスクを定期的に実行させたり、あるタスクを実行する時間を制限したりするために、通常はメインＣＰＵ１（図２７）に対してある一定周期で割り込み（図示せず）を入れる。（これをシステムクロックと呼ぶ）リアルタイムオペレーティングシステムは、システムクロック割り込みがある度に各タスクの状態を調べ、実行中のタスクを止めて別のタスクを実行させたりする。（これをスケジューリング、またはディスパッチと呼ぶ）また、リアルタイムオペレーティングシステムにおいては、各タスクに優先順位（実行の優先度）づけを行なう。この優先順位の意味は、より低い優先順位のタスクを実行中に、より高い優先順位のタスクを実行する準備ができたときは、その優先順位の低いタスクの実行を中断させて優先順位の高いタスクを実行させることをいう。（これを横取りという）

【０００８】図３０は、各タスク動作の時間的関係を示すタイムチャートである。縦軸は各処理の実行状況（実行または停止）を表し、横軸は時間の経過を表す。なお、図面作成の都合上、各処理としては、リアルタイムオペレーティングシステムと，割り込み処理と，（優先順位が高い順に）サーボ処理タスク，補正計算処理タスク，表示設定処理タスクを用いるものとする。Ｐ１は、リアルタイムオペレーティングシステムが動作している最中に割り込みが発生して制御が割り込み処理に移ったことを示す。Ｐ２は、割り込み処理が終了した後、リアルタイムオペレーティングシステムのスケジューラに制御が返る様子を示す。なお、スケジューラは、リアルタイムオペレーティングシステムでスケジューリングを行なう部分である。Ｐ３は、リアルタイムオペレーティングシステムのスケジューラにより、次に実行するタスクとしてサーボ処理タスクが選ばれたことを示す。Ｐ４は、サーボ処理タスクが終了した後、リアルタイムオペレーティングシステムに制御が返る様子を示す。Ｐ５は、リアルタイムオペレーティングシステムのスケジューリングによって、次に実行するタスクとして補正計算処理タスクが選ばれたことを示す。Ｐ６は、補正計算処理タスクが終了した後、リアルタイムオペレーティングシステムに制御が返る様子を示す。Ｐ７は、より優先順位の高いタスクを実行する準備ができていないので、優先順位が最も低い表示処理設定処理タスクに移ったことを示す。Ｐ８は、割り込みが発生したことで表示処理設定処理タスクを中断し、割り込み処理に制御が移ったことを示す。Ｐ９は、割り込み処理を終了した後、リアルタイムオペレーティングシステムのスケジューラに制御が返る様子を示す。（一般に、割り込み処理を終了した後は、もとのタスクに制御は戻らず、オペレーティングシステムのスケジューラに制御が返る。）Ｐ１０は、再び表示設定処理タスクに制御が移ったことを示す。Ｐ１１は、システムクロック（図示せず）によってリアルタイムオペレーティングシステムのスケジューラに制御が返ったことを示す。Ｐ１２は、サーボ処理タスクを実行する準備ができたので、サーボ処理タスクに移ったことを示す。（これは、表示設定タスクがサーボ処理タスクに実行権を横取りされた例である）Ｐ１３は、サーボ処理タスクはその実行を終了した後、自主的に実行権を放棄して、リアルタイムオペレーティングシステムのスケジューラに制御を返したことを示す。Ｐ１４は、再び表示設定処理タスクに制御が移ったことを示す。以降、各タスクは同様の動作を繰り返して、処理を続ける。

【０００９】なお、図３０に示したように、システムクロック以外の割り込みは不定期に発生するため、数値制御装置側ではその発生を予想できない。そこで、各タスクの特性に応じて、処理時間の配分を行なう必要がある。例えば、サーボ処理タスクは、前もって所定の単位時間のサーボモータの一回分の移動量を計算しておくために、所定の単位時間以内に次の移動量を計算しておく必要がある。つまり、少なくとも所定の単位時間以内には繰り返して実行する必要がある。一方、補正計算処理タスクは、例えば１回の実行で、サーボ処理タスクを３回だけ実行するために必要なデータを作成できるので、単位時間３回につき１回だけ実行すれば良い。また、表示設定処理タスクは他の処理がない時だけ実行すれば足る。

【００１０】以上で述べたように、数値制御装置５０１は、リアルタイムオペレーティングシステム上で制御ソフトウェアを実行したときに起動される複数のタスクの共同作業によって数値制御装置としての機能を実現する。

【００１１】さて、現在、数値制御装置の制御ソフトウェアの開発マシンとしては、ワークステーションが主流である。そこで、図３１に、ワークステーションを用いた数値制御装置の制御ソフトウェア開発のシステム構成

を例示する。１００は、コンパイラ，数値制御装置の制御ソフトウェアのソースコードやオブジェクトコードなど、各種ファイルを格納しているハードディスクである。１０１は、各種ファイルを一括管理したり、クライアントワークステーションの要求に応じてファイルを転送したりするファイルサーバである。１０２は、他のネットワークと通信するためのゲートウェイマシンである。１０３は、各種Ｉ／Ｏデバイスを制御するためのＩ／Ｏサーバである。１０４は、他のシステムに繋がっているネットワークである。１０５は、システム内を結ぶローカルネットワークである。１０６～１０８は、クライアントワークステーションである。

【００１２】次に、図３２に示す流れ図を用いて、数値制御装置の制御ソフトウェアを開発する手順を説明する。まず、ステップ１１１では、ファイルサーバ１０１からテキストファイルの編集を行なうエディターをローカルネットワーク１０５を通じてクライアントワークステーション１０６～１０８へダウンロードする。そして、エディターを使ってソースコードを作成する。一般に、複数の人間がそれぞれクライアントワークステーション１０６～１０８（図３１）を用いて、機能ごとにソースコードを作成する。

【００１３】ステップ１１２では、ファイルサーバ１０１からコンパイルを行なうコンパイラをローカルネットワーク１０５を通じてクライアントワークステーション１０６～１０８へダウンロードする。そして、コンパイラを使ってソースコードをコンパイルして、数値制御装置で実行可能なオブジェクトコードを作成する。なお、一般に、数値制御装置のメインＣＰＵとクライアントワークステーション１０６～１０８のＣＰＵとは異なるので、コンパイラとしては、クロスコンパイラを用いる。

【００１４】ステップ１１３では、コンパイルエラー（コンパイルしたときのエラー）が発生したか否か判定する。コンパイルエラーが発生したらステップ１１４に進み、コンパイルエラーが発生しなければステップ１１５に進む。ステップ１１４では、ソースコードを修正して、ステップ１１２に戻る。

【００１５】ステップ１１５では、ファイルサーバ１０１からリンクを行なうためのリンカとロケートを行なうためのロケータをローカルネットワーク１０５を通じてクライアントワークステーション１０６～１０８へダウンロードする。そして、リンカとロケータを用いてリンクとロケートを行ない、ロードモジュールを作成する。リンクは、複数のオブジェクトを１つの機能にまとめ、さらにそれらを結びつけて、モジュールを生成する作業である。ロケートは、モジュールをロケートする数値制御装置のメモリのアドレスを決定する作業である。ステップ１１６では、ロードモジュールを、他の媒体へコピーする。媒体としては、フロッピーディスク（数値制御装置がフロッピーディスクドライブを備えているとき）や、ＲＯＭがある。

【００１６】ステップ１１８では、数値制御装置は、媒体からロードモジュールを読み出して、メモリへロードする。ステップ１１８では、数値制御装置上でロードモジュールを実行し、意図した通りに動作するか否かを確認する。意図した通りに動作すれば処理を終了し、意図した通りに動作しなければ、ステップ１１４に戻る。ステップ１１８，１１９，１１４の一連の処理をデバッグと呼ぶ。

【００１７】ここで、ワークステーションおよびワークステーションのオペレーティングシステムについて説明する。ワークステーションには、以下に示すような特徴

ワークステーションにおいては多くのユーザが同時に使用して、その各ユーザに対して公平なオペレーティングシステムのサービスが要求される。つまり、ユーザによる優先順位のようなものも普通は存在しないし、また必ず決まった時間以内に処理しなくてはならない処理は、ほどんど無い。

そのため、ワークステーションのオペレーティングシステムとしては、多数の利用者が端末と対話しながら処理を行えるタイムシェアリングオペレーティングシステムを用いることが多い。

【００１８】タイムシェアリングオペレーティングシステムにおいて、あるまとまった処理の単位はプロセスと呼ばれる。プロセスは、リアルタイムオペレーティングシステムにおけるタスクと類似している。但し、一般に、リアルタイムオペレーティングシステムのもとでの各タスクのアドレス空間は１つの同じものである（線形アドレス空間と呼ぶ）のに対して、タイムシェアリングオペレーティングシステムでは、プロセスごとに固有なアドレス空間を持っている。具体例で示せば、リアルタイムオペレーティングシステムでは、ＣＰＵがアクセスして出来るアドレス（これを論理アドレス空間と呼ぶ）が０番地から１００００番地まであったと想定すると、リアルタイムオペレーティングシステムが０番地から１００００番地を占め、タスク１が１００１番地から２００００番地を占め、タスク２が２００１番地から２８０００番地までを占める。タスク１の２００００番地もタスク２の２００００番地も同じところを指し、同じデータを示す。そこで、タスク間でアドレスをやり取りすることで、お互いのデータなどを参照できる。一方、タイムシェアリングオペレーティングシステムでは、プロセスごとに固有なアドレス空間を持っているので、プロセス１とプロセス２がどちらもアドレス０番地からアドレス１００００番地までを持っていると想定すると、プロセス１の２００００番地とプロセス２の２００００番地では実際のメモリ上の異なるところを指すので、そのデータも異なる。

【００１９】次に、プロセスごとに固有なアドレス空間の実現方法を説明する。図３３は、クライアントワークステーションのメモリ態様図である。１３０はクライアントワークステーション、１３１はＣＰＵ、１３２はアドレス変換機構、１３３は変換テーブル、１３４は主記憶（メモリ）、１３５はネットワーク、１３６はファイルサーバ、１３７はハードディスクである。

【００２０】さて、クライアントワークステーション１３０において、ＣＰＵ１３１で区別できるアドレス空間（論理アドレス空間）は、一般に、主記憶につけられた物理的な番地（物理アドレス空間）よりも大きい。そこで、仮想記憶の手法により、物理アドレス空間外の論理アドレスをハードディスクなどの２次記憶装置に割り当てることで、２次記憶装置をも主記憶とみなせるようになる。つまり、主記憶の大きさが実装容量に制限されなくなるので、アドレス空間の制限を気にせずにソフトウェアを作成できるようになる。

【００２１】次に、仮想記憶の手法をさらに詳しく説明する。図３３において、ＣＰＵ１３１はアドレス変換機構１３２に対して論理アドレスを送る。アドレス変換機構１３２では、与えられた論理アドレスの一部または全部を取り出して変換テーブル１３３の配列の何番目かを示すインデックスとして用いる。変換テーブル１３３の内容が主記憶１３４のアドレスを示していれば主記憶１３４に対して所定の動作を行なう。また、変換テーブル１３３の内容がハードディスク１３７のブロック番号などを示していればファイルサーバ１３６に対してブロック番号に対応した内容を要求し、ネットワーク１３５を介して内容を受け取る。主記憶１３４に未使用領域が有れば、受け取ったデータを未使用領域に格納し、データを格納した領域を示すように変換テーブル１３３の内容を書き換える。また、主記憶１３４に未使用領域が存在しなければ、主記憶１３４に未使用領域が無ければ、主記憶１３４の既使用領域を選び、その内容をハードディスク１３７に書き戻し、その領域を新たに使用する。なお、主記憶１３４から変換テーブル１３３を逆にたどるソフトウェアの機能も備えられている。（これをコアマップと呼ぶ）

【００２２】さらに、プロセスごとに変換テーブルを持つことにより、各プロセスが同じ論理アドレス空間を利用することが出来る。この様子を図３４に示す。１４０はプロセス１、１４１はプロセス２、１４２はプロセス１の変換テーブル、１４３はプロセス２の変換テーブル、１４４は実メモリを示す。いま、プロセス１（１４０）とプロセス２（１４１）のいずれも、アドレス０番地（論理アドレスの０番地）をアクセスする場合を想定する。このとき、プロセス１（１４０）は、変換テーブル１４２の０番地の部分を指している。また、プロセス２（１４１）は、変換テーブル１４３の０番地の部分を指している。ところで、変換テーブル１４２の０番地と変換テーブル１４３の０番地は、実メモリ１４４上の異なる領域を指している。つまり、各プロセスが論理アドレス空間をフルに利用することが出来る。

【００２３】各プロセスは、例えば図３５に示すように、論理アドレス空間をいくつかの領域に分けて使用する。１５５は、１つのプロセスのすべての論理アドレス空間を示す。１５６は、そのプロセスのテキスト（プログラム）領域を示す。１５８は、予備領域である。この予備領域は、普段は未使用領域であるが、データ領域やスタック領域が不足したときには、図示の矢印の方向で、データ領域やスタック領域として用いられる。１５７は、そのプロセスが使用するデータ領域を示す。１５９は、スタック領域を示す。

【００２４】図３６は、タイムシェアリングオペレーティングシステムにおけるプロセスモデルを示す。２１０は計算機、２１１はプロセス、２１２は実体（例えばテキストコード）、２１３は実体において現在実行中のコードを示すプログラムカウンタである。各プロセスは、アドレス空間や、そのプログラムカウンタや、スタックや、レジスタ群などを持っている。なお複数のプロセス間で協調して処理を行なうときには、プロセス間通信の手段を用いて相互に連絡をとる必要がある。

【００２５】上記のタイムシェアリングオペレーティングシステムでは、図３６に示したように各プロセスが固有のアドレス空間を持つので、プロセスごとの変換テーブルの書き換えなどに手間がかかり、プロセスの切り換えに時間がかかりやすい欠点がある。また、各プロセス内の実行を行なう実体は１個であるので、複雑な処理を行なうアプリケーションプログラムに対応しずらくなってきた。そのため、近年、１つのプロセス中に複数の実体を置き、各実体を共通のアドレス空間のもとであたかも別々のプロセスであるかのように動かせるマルチスレッドオペレーティングシステムが提案されている。

【００２６】図３７に、マルチスレッドオペレーティングシステムのモデルを示す。２１０は計算機、２１１はプロセス、２１２は実体、２１３はプログラムカウンタである。実体２１２ａを、スレッド（またはライトウェイトプロセス）と呼ぶ。このスレッドは、プロセスと同様に、固有のプログラムカウンタやスタックを備えており、他のプロセスとは独立して動く。１つのプロセス内の全てのスレッドは同じアドレス空間を持っているので、共通変数を共有することも容易である。つまり、図３６を用いて先に説明したタイムシェアリングオペレーティングシステムのプロセスモデルでは、各プロセスが１つずつしかスレッドを持てないのに対して、マルチスレッドオペレーティングシステムのプロセスモデルでは、１つのプロセスが複数のスレッドを持つことができる。

【００２７】一般的に、各プロセスが持っていなければならない情報には次のようなものがある。

・アドレス空間
・共通変数（大域変数）
・開いているファイルの情報
・自分が生成したプロセスの情報
・タイマー情報
・シグナル情報
・セマフォーの情報
一方、スレッド毎に持っているべき情報は次のようなものがある。
・プログラムカウンタ
・スタック
・レジスタセット
・自分が生成したスレッドの情報
・自分自身の状態
【００２８】１つのプロセス中のすべてのスレッドは、プロセスごとに持つべき上記の情報を共有している。一方、スレッドごとに持つべき上記の情報は、各スレッドが個々に持っている。例えば図３８に示すように、１つのプロセスのすべての論理アドレス空間１５５は、テキスト領域１５６と，データ領域１５７と，予備領域１５８と；スレッド１のスタック領域２２０と，スレッド２のスタック領域２２１と，スレッド３のスタック領域２２２と，スレッド「Ｎ－１」のスタック領域２２３と，スレッドＮのスタック領域２２４とに分れている。つまり、１つのプロセス内の各スレッドが個別に持たねばならない領域のみを別個に割り当てることによって、１つのプロセス内に複数のスレッドを構成できる。
【００２９】次に、マルチスレッドオペレーティングシステムにおけるスケジューリングを説明する。図３０を用いて先に説明したリアルタイムオペレーティングシステムではタスクごとにスケジューリング行なったのに対して、マルチスレッドオペレーティングシステムでは、スレッドごとにスケジューリングを行なう。但し、マルチＣＰＵのときにはＣＰＵごとにスレッドが割り当てられるので、割り当てられたスレッドごとのスケジューリングを行なえばよい。
【００３０】
【発明が解決しようとする課題】上記従来の数値制御装置５０１では、数値制御装置としての機能を実現するタスクは、数値制御装置のリアルタイムオペレーティングシステムの下でしか走らせられない。そのため、数値制御装置にロードして制御ソフトウェアを実行させなければならず、面倒である。この発明は、上記問題点を解決するためになされたもので、数値制御装置としての機能を実現するタスクを汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステム上で実行できる数値制御装置の制御ソフトウェア実行システムを得ることを目的とする。
【００３１】
【課題を解決するための手段】請求項１の発明は、リアルタイムオペレーティングシステム上で制御ソフトウェアを実行したときに起動される複数のタスクの共同作業によって数値制御装置としての機能を実現する数値制御装置のタスクの機能を、汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステム上で実行するプロセスまたはタスク中のスレッドに割り当てると共に、スレッド間のスケジューリングの制御用に１つのスレッドを割り当てることを特徴とする数値制御装置の制御ソフトウェア実行システムを提供する。
【００３２】請求項２の発明は、請求項１の数値制御装置の制御ソフトウェア実行システムにおいて、汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステムを用いた計算機と、その計算機に通信回線を介して接続された数値制御装置とを具備してなり、前記計算機と前記数値制御装置の間に分散型共有メモリの機能を持たせ、前記数値制御装置のタスクの機能を、前記分散型共有メモリの機能を利用して、前記計算機と前記数値制御装置とで分散して処理することを特徴とする数値制御装置の制御ソフトウェア実行システムを提供する。
【００３３】請求項３の発明は、請求項１または請求項２の数値制御装置の制御ソフトウェア実行システムにおいて、汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステムを用いた計算機と、その計算機に通信回線を介して接続された数値制御装置とを具備してなり、前記計算機は前記数値制御装置への割り込み要求の有無を監視することを特徴とする数値制御装置の制御ソフトウェア実行システムを提供する。
【００３４】請求項４の発明は、請求項１から請求項３のいずれかの数値制御装置の制御ソフトウェア実行システムにおいて、複数のＣＰＵカードを有するＣＰＵプールを具備し、そのＣＰＵカード中のＣＰＵを前記数値制御装置のＣＰＵとして割り当てることを特徴とする数値制御装置の制御ソフトウェア実行システムを提供する。
【００３５】請求項５の発明は、請求項４の数値制御装置の制御ソフトウェア実行システムにおいて、前記ＣＰＵカード中のＣＰＵを前記複数の数値制御装置のＣＰＵとして割り当てることを特徴とする数値制御装置の制御ソフトウェア実行システムを提供する。
【００３６】請求項６の発明は、請求項５の数値制御装置の制御ソフトウェア実行システムにおいて、前記ＣＰＵプール中の１つのＣＰＵカードを割り込み検査専用のＣＰＵカードとして使用することを特徴とする数値制御装置の制御ソフトウェア実行システムを提供する。
【００３７】
【作用】請求項１の制御ソフトウェア実行システムでは、汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステム上で数値制

御装置の制御ソフトウェアを実行できるようになる。

【００３８】そこで、計算機で数値制御装置の制御ソフトウェアの動作を検証しやすくなり、数値制御装置の制御ソフトウェアの信頼性を向上できる。

【００３９】また、請求項１の制御ソフトウェア実行システムでは、数値制御装置の制御ソフトウェアのスケジューリングの制御用に１つのスレッドを生成する。これにより、計算機の汎用のマルチスレッドオペレーティングシステムがスレッド間のスケジューリング機能を備えていなくても、汎用のマルチスレッドオペレーティングシステムまたはマルチタスクオペレーティングシステム上で数値制御装置の制御ソフトウェアを実行できるようになる。

【００４０】請求項２の制御ソフトウェア実行システムでは、計算機と数値制御装置とでタスクを分散して処理する。これにより、数値制御装置に低性能のＣＰＵを使用でき、低コストにして効率的に数値制御装置の制御ソフトウェアを実行することが出来る。

【００４１】また、請求項２の制御ソフトウェア実行システムでは、分散型共有メモリの機能を利用して計算機と数値制御装置とで分散処理する。これにより、１プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムを用いたときにも、計算機と数値制御装置との間でタスクを分散して処理できるので、数値制御装置に低性能のＣＰＵを使用でき、低コストにして効率的に数値制御装置の制御ソフトウェアを実行することが出来る。

【００４２】また、本来は数値制御装置が起動すべきタスクを計算機が処理することで、数値制御装置のＣＰＵを不要にできる。

【００４３】請求項３の制御ソフトウェア実行システムでは、計算機は数値制御装置への外部割り込み要求の有無を監視する。これにより、数値制御装置のＣＰＵを無くしても割り込み処理に対処でき、数値制御装置の構成を簡略化できる。

【００４４】請求項４の制御ソフトウェア実行システムでは、複数のＣＰＵカードを有するＣＰＵプールを備える。これにより、数値制御装置の制御ソフトウェアの処理負荷に応じてＣＰＵを獲得・解放することで処理能力を調整できる。

【００４５】請求項５の制御ソフトウェア実行システムでは、ＣＰＵプールの中のＣＰＵを複数の数値制御装置のＣＰＵとして割り当てる。これにより、複数の数値制御装置の制御ソフトウェアを同時に実行できる。

【００４６】請求項６の制御ソフトウェア実行システムでは、ＣＰＵプールの中の１つのＣＰＵカードを割り込み検査専用のＣＰＵとして使う。これにより、外部割り込み信号に迅速に対処することが出来る。

【００４７】

【実施例】以下、図に示す実施例によりこの発明をさらに詳しく説明する。なお、これによりこの発明が限定されるものではない。

【００４８】第１実施例.

図１は、この発明の第１実施例の数値制御装置の制御ソフトウェア実行システムの全体構成図である。この制御ソフトウェア実行システムＳ１は、数値制御装置２００ａ１～２００ａｎと、ホスト計算機２０１と、ハードディスク２０２と、表示装置２０３と、キーボード２０４とを備える。数値制御装置２００ａ１～２００ａｎは、リアルタイムオペレーティングシステムのもとで制御ソフトウェアを実行して、数値制御装置としての機能を実現する。この数値制御装置２００ａ１～２００ａｎのハードウェア構成は、図２７を用いて先に説明した従来の数値制御装置５０１と同じである。ホスト計算機２０１は、汎用のマルチスレッドオペレーティングシステムのもとで数値制御装置の制御ソフトウェアを実行する。ハードディスク２０２は、ホスト計算機２０１の外部記憶装置である。表示装置２０３は、ユーザが必要とする各種の情報を表示する。キーボード２０４は、ユーザからの指示などを受け付ける。外部ネットワーク２０６は、ホスト計算機２０１を、汎用のワークステーション（図示せず）などと結ぶ。

【００４９】次に、ホスト計算機２０１により、数値制御装置の制御ソフトウェアを実行する手順を説明する。なお、説明の都合上、数値制御装置の制御ソフトウェアは、予めハードディスク２０２に格納されているものとする。まず、ユーザは、キーボード２０４を用いて数値制御装置の制御ソフトウェアを実現するプロセスの起動をホスト計算機２０１に指示する。ホスト計算機２０１のマルチスレッドオペレーティングシステムは、数値制御装置の制御ソフトウェアを実現するためのプロセスをハードディスク２０２から読み出し、プロセス管理テーブルに登録する。

【００５０】プロセス管理テーブルは、概念的には、図３３と図３４を用いて先に説明した変換テーブルと類似している。具体例で説明すれば、プロセス管理テーブルは、変換テーブル１３３（図３３）の配列の１番目からＭ番目までを、図３８に示したテキスト領域１５６を示すためのスロットとして確保し、変換テーブル１３３の配列の「Ｍ＋１」番目からＮ番目までを、図３８に示したデータ領域１５７を示すためのスロットとして確保したものである。

【００５１】次に、ホスト計算機２０１は、数値制御装置の制御ソフトウェアの実行状況に伴って、図３８に示したプロセスの論理アドレス空間１５５内の各領域を順にハードディスク２０２から読み出して内部メモリ（図示せず）へローディングする。

【００５２】次に、ホスト計算機２０１は、数値制御装置の制御ソフトウェアを実行したときのタスクに相当するプロセスを生成する。図２に、プロセスを生成するた

めの流れ図を示す。まず、ステップ２４１では、マルチスレッドオペレーティングシステムに備えられているシステムコールを利用して、数値制御装置としての１機能を実現している１タスク（数値制御装置のリアルタイムオペレーティングシステムのもとでの１タスク）と同じ処理を行なう１スレッドを生成する。具体的には、プロセス管理テーブルに新たなスレッドのスタック領域用のスロットを確保し、マルチスレッドオペレーティングシステムが認識するスレッドの表の中に新たなスレッドを追加する。

【００５３】ステップ２４２では、ステップ２４１で生成したスレッドに対応した処理を行なう。例えば、数値制御装置のリアルタイムオペレーティングシステムに対するシステムコールを、ホスト計算機２０１のマルチタスクオペレーティングシステムのスレッドに対するシステムコールに書き換える。なお、変換ライブラリを用いてシステムコールを変換するならば、書き換え負担を軽減できる。

【００５４】ステップ２４３では、数値制御装置のタスクに相当するスレッドをすべて生成したか否か判定する。すべてのスレッドを生成したならばステップ２４４に進み、生成すべきスレッドが残っていればステップ２４１に戻る。なお、すべてのスレッドを生成した時点では、ホスト計算機２０１の内部には、図３の概念図に示すように、マルチスレッドオペレーティングシステム２４０上に、数値制御装置のタスクと同じ機能を実現するスレッド１（２３１）～スレッド６（２３６）を含むプロセス２１１が構成されることになる。

【００５５】ステップ２４４では、スレッド間のスケージューリングを行なうスケージューリング方式を決定する。例えば、優先順位に応じて各スレッドを走らせるスケジューリング機能をホスト計算機２０１のマルチスレッドオペレーティングシステムが備えている場合には、システムコールを用いて各スレッドの優先順位を登録する。又は、ホスト計算機２０１のマルチスレッドオペレーティングシステムは、準備ができた順序でスレッドを走らせるスケジューラを有していることもある。ここで、“走る”とは、そのスレッドを構成するテキスト領域が実メモリ上に存在し、そのテキスト領域をＣＰＵが処理している状態のことである。

【００５６】図４は、スケジューラによるスケジューリングの一例を示す概念図である。２１１は、ホスト計算機２０１が実行するプロセスである。２３１～２３４は、数値制御装置のタスクと同じ機能を実現するスレッドである。２４０は、マルチスレッドオペレーティングシステムである。２５０は、準備が出来たスレッドの中から１つのスレッドを選び順に走らせるスケジューラである。２５１は、他のプロセスである。

【００５７】ステップ２４５では、数値制御装置の制御ソフトウェアが最初に初期化タスクを起動し、その初期化タスクが他のタスクを起動するような方式であるときは、初期化タスクと同じ処理を行なうスレッドをシステムコールにより起動する。具体的には、マルチスレッドオペレーティングシステムのスケジューリング対象のスレッドの表の中に、初期化タスクと同じ処理を行なうスレッドを加えればよい。これらの処理により、各スレッドは、予め決定された順序に応じて走ることになる。なお、数値制御装置の制御ソフトウェアが上記のような方式でない場合には、従来の数値制御装置の制御ソフトウェアと同じ順序で該当スレッドを起動すればよい。

【００５８】上記ではホスト計算機２０１のマルチスレッドオペレーティングシステムの機能を利用してスレッド間のスケジューリングを行なった。しかし、マルチスレッドオペレーティングシステムがスレッド間のスケジューリング機能を備えていない場合には上記の対応はできない。

【００５９】そこで、この場合は、スレッド間のスケジューリングの制御用に１つのスレッドを生成すればよい。

【００６０】つまり、ホスト計算機２０１は、数値制御装置における各タスクをスレッドとするのに加えて、スケジューラ２５０（図４）と同じ働きをするスケジューラスレッドを生成する。

【００６１】図５は、ホスト計算機２０１により生成されたスケジューラスレッドによるスケジューリングの概念図である。これは、図４を用いて先に説明したスケジューリングの概念に、スケジューラスレッド２５２を加えたものである。そして、スケジューラスレッド２５２は、マルチスレッドオペレーティングシステム２４０上のスケジューラ２５０から数値制御装置の制御ソフトウェアを実行するプロセス２１１の実行権を獲得し、スレッド１（２３１）～スレッド４（２３４）のスケジューリングをすべて行なう。

【００６２】以上の第１実施例の数値制御装置の制御ソフトウェア実行システムによれば、数値制御装置の制御ソフトウェアのスケジューリングの制御用に１つのスレッド（スケジューラスレッド）を生成するので、ホスト計算機の汎用のマルチスレッドオペレーティングシステムがスレッド間のスケジューリング機能を備えていなくても、ホスト計算機における汎用のマルチスレッドオペレーティングシステムのもとで数値制御装置の制御ソフトウェアを実行させることができるようになる。そのため、制御ソフトウェア開発時のデバッグでは、ホスト計算機上で開発された制御ソフトウェアをいちいち数値制御装置にロードして動作を確認する必要がなくなり、開発期間を短縮できる。また、ホスト計算機上に構築されたＣＡＤ／ＣＡＭシステムに数値制御装置のシミュレータ機能を組み込む際には、数値制御装置の制御ソフトウェアを汎用のマルチスレッドオペレーティングシステム上で動くように組み直さなくてよいので、手間がかから

ない。また、ホスト計算機と数値制御装置のオペレーティングシステム等の違いを考慮したスケジューリングが不要となるので、数値制御装置の既存のソフトウェアを短期間で移植できる。

【００６３】なお、上記ではホスト計算機２０１上で動作する汎用のマルチスレッドオペレーティングシステムのもとで数値制御装置の制御ソフトウェアのタスクに相当するスレッドを走らせたのに対して、数値制御装置に汎用のマルチスレッドオペレーティングシステムを動作させ、そのもとで数値制御装置の制御ソフトウェアのタスクに相当するスレッドを走らせてもよい。つまり、数値制御装置は、上記のようにホスト計算機２０１が行なったのと同様にして、マルチスレッドオペレーティングシステム上のスレッドとして、数値制御装置の制御ソフトウェアを実行する。

【００６４】このようにすれば、数値制御装置の制御ソフトウェアを汎用のマルチスレッドオペレーティングシステム上で動作するソフトウェアとして開発できるので、ホスト計算機で数値制御装置の制御ソフトウェアの動作を検証しやすくなり、数値制御装置の制御ソフトウェアの信頼性を向上できる。

【００６５】第２実施例.

この発明の第２実施例の数値制御装置の制御ソフトウェア実行システムは、前記第１実施例の構成に加えて、ホスト計算機と、１つの数値制御装置との間でスレッド間通信を行なわせ且つ分散型共有メモリの機能を利用して、上述のスレッドを分散して走らせるものである。

【００６６】図６は、第２実施例の数値制御装置の制御ソフトウェア実行システムＳ３の全体構成図である。この制御ソフトウェア実行システムＳ３は、第１実施例の制御ソフトウェア実行システムＳ１（図１）とほぼ同じ構成である。但し、第１実施例では数値制御装置２００ａ１～２００ａｎがホスト計算機２０１とは孤立していたのに対して、第２実施例では数値制御装置２００ｂ１～２００ｂｎとホスト計算機２０１とを内部ネットワーク２０５を介して接続した点が異なる。

【００６７】図７は、数値制御装置２００ｂ１のハードウェア構成図である。なお、数値制御装置２００ｂ２～２００ｂｎも同じ構成である。１は、各部の作動を制御したり、数値制御に必要な演算を行なったりするメインＣＰＵである。２は、各部を結ぶシステムバスである。３は、数値制御装置の主要機能を実現する制御ソフトウェアなどを格納するＲＯＭ（不揮発性の記憶装置）である。４は、一時記憶やワークエリアなどに用いるＲＡＭ（揮発性の記憶装置）である。５は、外部との間でシリアル通信によりデータのやり取りを行なうデータＳＩＯインタフェース部である。６は、数値制御装置の運転状態を表示したり、数値制御装置に与えた指令を確認したりするための表示装置である。７は、数値制御装置に指令を与えるためのキーボード（入力装置）である。８は、サーボモータを制御するための指令を演算するサーボ制御部である。９は、サーボ制御部８から受け取った指令を増幅してサーボモータへ駆動信号を出力するサーボアンプである。１０は、工作機械（図示せず）の加工部を制御するためのサーボモータである。１１は、工作機械との間で、サーボ制御指令以外のデータをやり取りするためのプログラムコントローラ部である。１２は、メインＣＰＵ１に与えられる、システムクロック（図示せず）と外部割り込み信号を表している。システムクロックは、数値制御装置全体を制御するためのクロック信号である。また、外部割り込み信号は、電源異常や非常停止などのイベント（緊急な出来事）の発生をメインＣＰＵ１に通知するための信号である。１３２は、図３３を用いて先に説明した仮想記憶を実現するためのアドレス変換機構である。２０７は、内部ネットワーク２０５を通じてホスト計算機２０１や他の数値制御装置２００ｂ２～２００ｂｎとの間で通信を行なうためのネットワークインタフェースである。

【００６８】上記制御ソフトウェア実行システムＳ３では、ホスト計算機２０１上に構築された汎用のＣＡＤ／ＣＡＭシステムによりＣＡＤ／ＣＡＭデータを生成し、そのＣＡＤ／ＣＡＭデータを数値制御装置の加工プログラムに変換して、その加工プログラムに基づいて加工を行なわせることが出来る。図８は、ホスト計算機２０１上に構築された汎用のＣＡＤ／ＣＡＭシステムの動作の概念図である。ホスト計算機２０１は、ＣＡＤ／ＣＡＭプロセス３４０と，データ変換プロセス３４１と，数値制御装置の制御ソフトウェアを実現する数値制御装置プロセス３４２とを生成して起動する。そして、ＣＡＤ／ＣＡＭプロセス３４０にはデータ出力用スレッド３４３を走らせておく。ＣＡＤ／ＣＡＭプロセス３４０は通常の動作によりＣＡＤ／ＣＡＭデータを作成する。データ出力用スレッド３４３は、ＣＡＤ／ＣＡＭデータをデータ変換プロセス３４１のスレッド３４４に入力する。データ変換プロセス３４１では、内部のスレッドによりＣＡＤ／ＣＡＭデータを数値制御装置の入力データに変換し、データ出力スレッド３４５に渡す。データ出力スレッド３４５は、数値制御装置の入力データを数値制御装置プロセス３４２のデータ入力（設定）スレッド３４６に送る。数値制御装置プロセス３４２は、数値制御装置の入力データを加工プログラムに変換して、内部ネットワーク２０５を通じて数値制御装置に入力し、加工を行なわせる。

【００６９】上記のようにホスト計算機上のＣＡＤ／ＣＡＭシステムにより生成されたＣＡＤ／ＣＡＭデータから数値制御装置の加工プログラムを自動作成すれば、加工プログラムを作成するためのユーザ負担を軽減できる。

【００７０】次に、スレッドを分散して走らせることについて説明する。

【００７１】スレッドの分散の具体例を説明すれば、ある数値制御装置が低性能で安価なメインＣＰＵを有しており、ホスト計算機２０１が高性能で高価なＣＰＵを有しているときには、簡単な処理を行なうスレッドは数値制御装置に走らせて、複雑な処理を行なうスレッドはホスト計算機２０１に走らせるのが好適である。又、別の具体例を説明すれば、図２８を用いて先に説明した補間処理２４やサーボ処理２５を行なうスレッドなどは数値制御装置に走らせ、加工プログラム入力処理２１や補正計算処理２２を行なうスレッドはホスト計算機２０１に走らせるのが好適である。

【００７２】図９は、数値制御装置２００ｂ１とホスト計算機２０１との間で行なわれるスレッド間通信の説明図である。２１１は、プロセスである。２３１はスレッド１、２３２はスレッド２、２３３はスレッド３、２３４はスレッド４である。２６０はノード１、２６１はノード２である。ここで、ノードとは、ネットワークの枝の端点にあたる情報処理機器の総称である。例えば、ノード１（２６０）はホスト計算機２０１、ノード２（２６１）は数値制御装置２００ｂ１である。２６２は、オペレーティングシステム２４０ａのもとでスレッド１とスレッド２との間の通信を処理するスレッド間通信処理部である。２６３は、ノード１とノード２とを結ぶノード間ネットワークである。

【００７３】いま、ノード１（２６０）内のスレッド１（２３１）からノード２（２６１）内のスレッド３（２３３）にメッセージを送る場合を想定する。スレッド１（２３１）はシステムコールによって相手スレッドを指定し、メッセージを送る要求をオペレーティングシステム２４０ａに対して発行する。このシステムコールによって制御は、オペレーティングシステム２４０ａに移る。システムコールの内容を解析することで、制御はスレッド間通信処理部２６２に移る。スレッド間通信処理部２６２は、相手スレッドがあるノード（ノード２）を判定し、ノード２（２６１）に対してノード間ネットワーク２６３を介してメッセージを送る。そのメッセージは、ノード２（２６１）のスレッド間通信処理部２６２に渡り、スレッド間通信処理部２６２は、指定されたスレッド３（２３３）に対してメッセージを渡す。

【００７４】上記の想定では、スレッド１（２３１）からスレッド３（２３３）に直接にメッセージを送るものとしたが、マルチスレッドオペレーティングシステムによってはメールボックスを用いてメッセージを渡すことも出来る。図１０は、ノード１（２６０）内のスレッドがノード２（２６１）内のスレッド３（２３３）にメールボックスを用いてメッセージを送る場合の概念図である。スレッド１（２３１）はシステムコールによって相手スレッドのメールボックスを指定し、メッセージを送る要求をオペレーティングシステム２４０ａに対して発行する。このシステムコールによって、制御はオペレーティングシステム２４０ａに移る。システムコールの内容を解析することで、制御はスレッド間通信処理部２６２に移る。スレッド間通信処理部２６２は指定されたメールボックスがあるノード（ノード２）を判定して、ノード２（２６１）に対してノード間ネットワーク２６３を通してメッセージを送る。そのメッセージはノード２（２６１）のスレッド間通信処理部２６２に渡る。スレッド間通信処理部２６２は、メールボックス２６４にメッセージを格納する。その後、スレッド間通信処理部２６２は、スレッド３（２３３）からポートを指定したメッセージ読み込み要求があったらメッセージを渡す。読み込み要求の方がメッセージの到着よりも速かった場合には、メッセージが到着するまでスレッド３（２３３）を待機させる。

【００７５】次に、分散共有メモリの機能を利用して、スレッドを分散して走らせることについて説明する。

【００７６】１プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムでは、分散型共有メモリの機能を利用して、計算機２０１と数値制御装置２００ｂ１とで分散して処理する。すなわち、ホスト計算機２０１に数値制御装置の制御ソフトウェアの機能を実現するプロセスを走らせ、１つの数値制御装置２００ｂ１にホスト計算機２０１のプロセスとは独立したスレッドを走らせる。その際、１プロセス内のスレッドでのアドレス空間の共有を、分散型共有メモリの機能により実現する。

【００７７】図１１は、分散型共有メモリの概念図である。ノード１（２６０）のＣＰＵ１３１が、ある論理アドレスに対応した主記憶１３４の内容を読むときは、アドレス変換機構１３２と変換テーブル１３３を用いて、主記憶１３４の物理アドレスを算出して、内容を得る。例えば、論理アドレスがノード１（２６０）の主記憶１３４に対応しているならば、ノード１（２６０）の変換テーブル１３３は主記憶１３４の物理アドレスを算出する。また、論理アドレスがノード２（２６１）の主記憶１３４に対応しているならば、ノード１（２６０）の変換テーブル１３３は、ノード間ネットワーク１３５を介してノード２（２６１）の変換テーブル１３３に主記憶１３４の内容の読出しを要求する。

【００７８】上記では、ホスト計算機２０１と１つの数値制御装置２００ｂ１とに分散してスレッドを走らせたのに対して、図１２に示すように、ホスト計算機２０１と複数の数値制御装置２００ｂ１～２００ｂｎとに分散してスレッドを走らせてもよい。

【００７９】例えば、実時間処理が必要な補間処理２４やサーボ処理２５（図２８）を行なうスレッド２３１は、それぞれの数値制御装置２００ｂ１～２００ｂｎで走らせ、実時間処理が不要な加工プログラム入力処理２１や補正計算処理２２（図２８）を行なうスレッド２３２は、ホスト計算機２０１上で共通に走らせるのが好適

である。なお、ホスト計算機２０１上のスレッド２３２は、数値制御装置２００ｂ１～２００ｂｎ対応で少なくとも１つずつ走らせる。

【００８０】以上の第２実施例の数値制御装置の制御ソフトウェア実行システムによれば、前記第１実施例の効果に加えて、１プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムを用いたときにも、ホスト計算機２０１と数値制御装置２００ｂとの間でタスクを分散して走らせるので、数値制御装置には低性能のＣＰＵを使用でき、低コストにして効率的に数値制御装置の制御ソフトウェアを実行することが出来る。

【００８１】上記では、キーボードと表示装置を備えた数値制御装置２００ｂ１～２００ｂｎを用いたが、キーボードと表示装置を備えていない数値制御装置を用いてもよい。図１３に示す制御ソフトウェア実行システムＳ７は、これまでの制御ソフトウェア実行システムと同様の構成であるが、キーボードと表示装置を備えていない数値制御装置２００ｅ１～２００ｅｎを用いた点が異なる。

【００８２】図１４は、数値制御装置２００ｅ１のハードウェア構成図である。なお、数値制御装置２００ｅ２～２００ｅｎも同じ構成である。数値制御装置２００ｅ１は、第２実施例の数値制御装置２００ｂ１（図７相当）から、表示装置６とキーボード７とを取り除いた構成である。

【００８３】図１５は、この制御ソフトウェア実行システムＳ７の動作を示す概念図である。数値制御装置２００ｅ１の表示設定処理を行なうスレッド２７３は、他のスレッド２７２から表示要求があったとき、自分では表示処理を行わず、ホスト計算機２０１の表示装置を制御するスレッド２７０に対して内部ネットワーク２０５を介して表示要求を送る。スレッド２７０は表示要求を受け取ると、表示装置２０３に対して表示処理を行なう。つまり、数値制御装置２００ｅ１の表示装置に表示する代りに、ホスト計算機２０１の表示装置２０３に表示する。（そのときは、ホスト計算機本来の画面を表示できなくなるので、表示装置２０３はホスト計算機本来の画面または１つの数値制御装置のいずれかの画面を表示することになる）また、ホスト計算機２０１のキーボード２０４が数値制御装置２００ｅ１への指示を受け付けると、スレッド２７１は、その指示内容を内部ネットワーク２０５を介して数値制御装置２００ｅ１の表示設定処理を行なうスレッド２７３へ送る。つまり、数値制御装置２００ｅ１のキーボードを用いて入力する代りに、ホスト計算機２０１のキーボード２０４を用いて入力する。

【００８４】上記によれば、ホスト計算機２０１の表示装置２０３に１つの数値制御装置の画面を表示すると共に、ホスト計算機２０１のキーボード２０４を用いて１つの数値制御装置への指示を入力できるようになるので、数値制御装置の表示装置やキーボードがいらなくなり、数値制御装置の構成を簡略化して安価にすることが出来る。

【００８５】ホスト計算機２０１の表示装置としてビットマップディスプレイを用いてもよい。図１６は、ビットマップディスプレイを用いた制御ソフトウェア実行システムの全体構成図である。この制御ソフトウェア実行システムＳ８は、前記制御ソフトウェア実行システムＳ７とほぼ同じ構成である。但し、数値制御装置２００ｆ１～２００ｆｎを用いた点と、ホスト計算機２０１の表示装置２０３ａとしてビットマップディスプレイを用いた点が異なる。

【００８６】図１７は、この制御ソフトウェア実行システムＳ８における表示処理の概念図である。ホスト計算機２０１は、ホスト計算機本来の画面である画面１（２８０）と画面２（２８１）と画面３（２８２）に対応して、画面を制御するスレッド２８４，２８５，２８６を生成する。数値制御装置２００ｆ１では、表示要求があると、実施例７と同様にして、ホスト計算機２０１のスレッド２８４，２８５，２８６に対して内部ネットワーク２０５を介して表示要求を送る。そこで、ホスト計算機２０１は、ホスト計算機本来の画面と数値制御装置２００ｆ１の複数の画面を表示装置２０３ａに表示する。

【００８７】上記によれば、ホスト計算機の表示装置にホスト計算機本来の画面と１つの数値制御装置の複数の画面を１度に表示するので、ユーザは多彩な情報を１度に得ることができるようになり、各種の指示を入力する際の操作性が向上する。

【００８８】ホスト計算機２０１の表示装置２０３ａ（図１６相当）にホスト計算機本来の画面と複数の数値制御装置の画面を同時表示できるようにしてもよい。そのためには、図１７を用いて先に説明したホスト計算機２０１のスレッドを、表示対象の数値制御装置２００ｇ１～２００ｇｎの個数に応じてさらに増やせば良い。

【００８９】このようにすれば、ホスト計算機の表示装置に同時表示された画面からホスト計算機本来の情報と複数の数値制御装置の情報を１度に得ることが出来るため、操作性がさらに良くなる。

【００９０】マウスなどのポインティングデバイスを用いて任意の数値制御装置に対して指令を与えられるようにしてもよい。図１８は、マウスを備えた制御ソフトウェア実行システムの全体構成図である。この制御ソフトウェア実行システムＳ１０は、図１６とほぼ同じ構成であるが、マウスMなどのポインティングデバイスを用いて任意の数値制御装置２００ｈ１～２００ｈｎに対して指令を与えられる点が異なる。

【００９１】図１９は、この制御ソフトウェア実行システムＳ１０における表示処理の概念図である。ホスト計算機２０１は、図１７を用いて先に説明した画面を制御

するスレッドに加えて、スレッド２９２を生成する。このスレッド２９２は、ユーザにより操作されたマウスMなどのポインティングデバイスの動きを検知して、ポインティングデバイスカーソル２９１がどの画面にいるかを判断して、それに応じて例えば画面１（２８０）を制御するスレッド２９３に対してポインティングデバイスカーソル２９１を表示するように要求を出す。そのとき、キーボード２０４（図１８）からの入力は、画面１（２８０）への入力とみなして、入力情報を画面１（２８０）を制御するスレッド２９３に対して送る。スレッド２９３は、当該数値制御装置（図示せず）（のスレッド）に対して入力情報を送る。

【００９２】上記によれば、ユーザは、指示を与える数値制御装置をホスト計算機の画面上のポインティングデバイスカーソルにより選択できるので、複数の数値制御装置に指示を与える場合でもいちいち移動する必要がなくなる。

【００９３】第３実施例.

この発明の第３実施例の数値制御装置の制御ソフトウェア実行システムは、前記第１実施例の構成に加えて、数値制御装置２００ｉ１～２００ｉｎ（図示せず）に対する外部割り込み信号をホスト計算機で処理できるようにしたものである。

【００９４】図２０は、数値制御装置２００ｉ１のハードウェア構成図である。なお、数値制御装置２００ｉ２～２００ｉｎも同じ構成である。数値制御装置２００ｉ１は、第２実施例の数値制御装置２００ｂ１（図７）から、メインＣＰＵ１とＲＯＭ３と表示装置６とキーボード７とアドレス変換機構１３２を取り除き、割り込み信号処理部３００を追加した構成である。この構成は、ホスト計算機２０１のＣＰＵを数値制御装置２００ｉ１～２００ｉｎでも使用することで、数値制御装置２００のメインＣＰＵ１およびアドレス変換機構１３２（図７相当）を不要としたものである。

【００９５】次に、この制御ソフトウェア実行システムＳ１１の動作を説明する。なお、説明の都合上、数値制御装置２００ｉ１についての動作のみを説明するが、数値制御装置２００ｉ２～２００ｉｎについても同様である。外部割り込み信号１２は、数値制御装置２００ｉ１に対して電源異常や非常停止などのイベント（緊急な出来事）の処理を要求する時に発生され、数値制御装置２００ｉ１の割り込み信号処理部３００に伝えられる。このとき、割り込み信号処理部３００は、外部割り込み信号１２をＲＡＭ４などにラッチ（一時的に記憶）しておく。一方、ホスト計算機２０１は、ＲＡＭ４などの内容を定期的にチェックするスレッドを生成し、外部割り込み信号１２がラッチされていれば、数値制御装置２００ｉ１に対して何らかの要求があったと判定して、ホスト計算機２０１上の数値制御装置の制御ソフトウェアを実現するスレッド（群）に対してそれを知らせる。その後は、これまでの実施例と同様に動作する。

【００９６】以上の第３実施例の制御ソフトウェア実行システムによれば、ホスト計算機が数値制御装置への外部割り込み信号の有無を監視するので、数値制御装置のＣＰＵを無くしても割り込み処理に対処でき、数値制御装置の構成を簡略化できる。

【００９７】第４実施例.

図２１は、この発明の第４実施例の数値制御装置の制御ソフトウェア実行システムの全体構成図である。この制御ソフトウェア実行システムＳ１２は、ハードディスク２０２への読み書きを行なうファイルサーバ２０８と、ＣＰＵプール３０１と、外部ネットワーク２０６と接続したゲートウェイ部２０７０を備えている。また、数値制御装置数値制御装置２００ｊ１～２００ｊｎのハードウェア構成は、２００ｉ１～２００ｉｎ（図２０）と同じである。

【００９８】図２２は、ＣＰＵプール３０１の内部構成図である。ＣＰＵプール３０１のスロットには、複数のＣＰＵカード３０２が挿入されている。各ＣＰＵカード３０２の間は、バス３１２により結ばれている。これらＣＰＵカード３０２は、同種のＣＰＵに統一されていなくてもよい。各ＣＰＵカード３０２のＣＰＵ３１１，３１３～３２５は、それぞれローカルメモリ３１０，３１２～３２６を備えている。

【００９９】次に、この制御ソフトウェア実行システムＳ１２の動作を説明する。ＣＰＵプール３０１の１つのＣＰＵカード３０２にホスト計算機２０１のＣＰＵの役目を果たさせる。これをホスト機能ＣＰＵカードと呼ぶ。このホスト機能ＣＰＵカードが第３実施例と同様の処理により数値制御装置２００ｊ１～２００ｊｎにＣＰＵを割り当てる。例えば、図２２においてＣＰＵ３１１がホスト機能ＣＰＵであるとき、数値制御装置２００ｊ１のＣＰＵとしてＣＰＵ３１３を割り当てる。このとき、数値制御装置２００ｊ１の制御ソフトウェアを実行するスレッドは、ローカルメモリ３１０とローカルメモリ３１２（図２２）にローディングされて、それぞれホスト機能ＣＰＵのマルチスレッドオペレーティングシステムのもとで走る。これらの処理は、第２実施例で述べた処理と類似しているが、スレッド間通信はネットワークではなくバス３１２を通して行なう点が異なる。

【０１００】なお、ＣＰＵの能力が不足するとき（例えばホスト機能ＣＰＵのオペレーティングシステムのもとで走るスレッドの数がある一定量を越えたようなとき）は、ＣＰＵプール３０１（図２２）内のＣＰＵカード３０２に余裕がある限り新たなＣＰＵカード３０２を獲得することで、ＣＰＵの処理能力を調整できる。具体的には、ＣＰＵの処理能力を多く必要とする時には複数のＣＰＵカード３０２を使用し、ＣＰＵの処理能力がそれほど必要でない時には１つのＣＰＵカード３０２を使用すればよい。

【０１０１】図２３は、新たなＣＰＵカード３０２を獲得するときの動作を示す概念図である。３２０，３２１は、ホスト機能ＣＰＵ３１１上で走っているスレッドを示している。３２２～３２３は、数値制御装置２００ｊ１の制御ソフトウェアを実行するＣＰＵ３１３上で走っているスレッドである。

【０１０２】スレッド３２２は、ＣＰＵ３１２上で走っているスレッドの数を周期的に数える。スレッド３２０は、スレッド３２２と周期的にメッセージを交換することでＣＰＵ３１３上で走っているスレッドの数を認識する。スレッドの数が所定値を越えると、スレッド３２０は、数値制御装置２００ｈ１の処理を行なうＣＰＵ３１３を獲得したのと同じ手順で新たなＣＰＵカードを獲得する。ここでは、説明の都合上、図２２におけるＣＰＵ３２５を新たに獲得するものとする。まず、新たに走らせるべきスレッドのコードと、ＣＰＵ３２５のオペレーティングシステムをハードディスク２０２（図２１）から読み出し、ローカルメモリ３２６にローディングする。初期化スレッドを走らせた後、スレッド３２２に対して新たに獲得したＣＰＵ３２５についての情報を送る。次に、それまでＣＰＵ３１３で走らせていたスレッド３２２をＣＰＵ３２５に移して走らせる。

【０１０３】以上の第４実施例の制御ソフトウェア実行システムによれば、複数のＣＰＵカードを有するＣＰＵプールを備えるので、数値制御装置の制御ソフトウェアの処理負荷に応じてＣＰＵを獲得・解放することで処理能力を調整できる。

【０１０４】第５実施例.

この発明の第５実施例の数値制御装置の制御ソフトウェア実行システムは、第４実施例とほぼ同じである。但し、第４実施例では１度に１つずつの数値制御装置を制御するのに対して、第５実施例では１度に複数の数値制御装置を制御できる点が異なる。

【０１０５】この制御ソフトウェア実行システムＳ１３の動作は、第４実施例の制御ソフトウェア実行システムＳ１２の動作とほぼ同じである。但し、ＣＰＵプール３０１（図２１）中のホスト機能ＣＰＵカード上のスレッドが第４実施例と同様の処理により数値制御装置に順にＣＰＵを割り当てていく。

【０１０６】以上の第５実施例の制御ソフトウェア実行システムＳ１３によれば、ＣＰＵプールの中のＣＰＵを複数の数値制御装置のＣＰＵとして割り当てることができるので、第４実施例による効果に加えて、複数の数値制御装置の制御ソフトウェアを同時に実行できるようになる。

【０１０７】第６実施例.

この発明の第６実施例の数値制御装置の制御ソフトウェア実行システムＳ１４は、第５実施例とほぼ同じである。但し、第６実施例では、ＣＰＵプール３０１（図２２）中の１つのＣＰＵカード３０２を割り込み検査専用のＣＰＵカードとして使用する点が異なる。

【０１０８】この制御ソフトウェア実行システムＳ１４の動作は、第５実施例の制御ソフトウェア実行システムＳ１３の動作とほぼ同じである。但し、ＣＰＵカード３０２のローカルメモリへのコードのローディングはホスト機能ＣＰＵのスレッドが行ない、割り込みを発見した場合には、当該数値制御装置を制御しているＣＰＵカード３０２上のスレッドにメッセージを送ってそれを通知する。また、割り込みが当該数値制御装置に対する最初の割り込みであった場合には、それをホスト機能ＣＰＵのスレッドに送って、これまで述べたような手順でその数値制御装置の動作を開始させる。

【０１０９】以上の第６実施例の制御ソフトウェア実行システムによれば、ＣＰＵプールの中の１つのＣＰＵカードを割り込み検査専用のＣＰＵとして使うので、外部割り込み信号に迅速に対処することができるようになる。

【０１１０】図２４に示す制御ソフトウェア実行システムＳ１４’は、制御ソフトウェア実行システムＳ１４とほぼ同じ構成である。但し、ＣＰＵカード３０２のＣＰＵごとのローカルメモリ３１０，３１２～３２６（図２２相当）に代えて、ＣＰＵカード３０２のＣＰＵで共有できる共有メモリ３３０を設けた点と、数値制御装置２００ｋ１～２００ｋｎを用いた点とが異なる。

【０１１１】図２５は、数値制御装置２００ｋ１のハードウェア構成図である。なお、数値制御装置２００ｋ２～２００ｋｎは、数値制御装置２００ｋ１と同じ構成である。数値制御装置２００ｋ１は、実施例１１の数値制御装置２００ｉ１（図２０）からＲＡＭ４を取り除いた構成である。

【０１１２】この制御ソフトウェア実行システムＳ１４’の動作は、制御ソフトウェア実行システムＳ１４の動作とほぼ同じであるが、仮想的な分散型共有メモリを用いる代りに、実際に同じ物理アドレスでスレッドを走らせる。このときは、あるプロセス内のスレッドを任意のノード上で走らせることが出来る。

【０１１３】以上のようにすれば、ＣＰＵカードのＣＰＵごとにローカルメモリを備えなくても、外部割り込み信号に迅速に対処することができるようになる。

【０１１４】変形実施例.

数値制御装置２００ｐ１（図示せず）に汎用のマルチスレッドオペレーティングシステム自体とデータ入力スレッドだけを予め走らせておき、データ入力スレッドは、ホスト計算機２０１のデータ出力スレッドから、ホスト計算機２０１のハードディスク２０２内にファイルとして管理されている数値制御装置の制御ソフトウェアのコード部分を通信で受け取って、そのコードを数値制御装置２００ｐ１のＲＡＭ４に順に格納していく、というようにしてもよい。

【０１１５】このようにすれば、数値制御装置はホスト

計算機からダウンロードされた制御ソフトウェアを実行できるので、ホスト計算機上で開発された制御ソフトウェアを速やかに実行できるようになる。また、ホスト計算機は、数値制御装置の制御ソフトウェアをファイルとして管理するので、管理効率が向上する。

【０１１６】ダウンロード対象の数値制御装置を１台に限らず、複数台にしてもよい。

【０１１７】ダウンロード対象の数値制御装置を複数台とすれば、複数の種類の数値制御装置の制御ソフトウェアをファイルとして一括して管理できるため、管理効率がさらに向上する。

【０１１８】数値制御装置にフラッシュＲＯＭを追加してもよい。

【０１１９】図２６は、フラッシュＲＯＭを追加した数値制御装置２００ｑ１のハードウェア構成図である。なお、数値制御装置２００ｑ２～２００ｑｎも同じ構成である。数値制御装置２００ｑ１は、第２実施例の数値制御装置２００ｂ１（図７）に、フラッシュＲＯＭ１１３３を追加した構成である。

【０１２０】次に、この数値制御装置２００ｑ１に制御ソフトウェアをダウンロードするときの動作を説明する。数値制御装置２００ｑ１には、汎用のマルチスレッドオペレーティングシステム自体と，データ入力スレッドと，フラッシュＲＯＭ書き込みスレッドだけを予め走らせておく。そして、データ入力スレッドは、ホスト計算機２０１のデータ出力スレッドから、ホスト計算機２０１のハードディスク２０２内にファイルとして管理されている数値制御装置の制御ソフトウェアのコード部分を通信で受け取って、そのコードを順にフラッシュＲＯＭ書き込みスレッドに渡す。フラッシュＲＯＭ書き込みスレッドは、与えられたコードをフラッシュＲＯＭ１１３３に書き込んでいく。

【０１２１】上記によれば、数値制御装置の制御ソフトウェアをフラッシュＲＯＭに保持しておけるため、数値制御装置を立ち上げる度に制御ソフトウェアをダウンロードしなくてよい。また、フラッシュＲＯＭに書き込まれた内容は１度にまとめて消去できるので、制御ソフトウェアの書き換え時間を短縮できる。

【０１２２】なお、上記説明では、ホスト計算機２０１のオペレーティングシステムとして、汎用のマルチスレッドオペレーティングシステムを用いたが、汎用のマルチタスクオペレーティングシステムを用いても同等である。

【０１２３】

【発明の効果】請求項１の制御ソフトウェア実行システムによれば、数値制御装置の制御ソフトウェアのスケジューリングの制御用に１つのスレッドを生成するので、計算機の汎用のマルチスレッドオペレーティングシステムがスレッド間のスケジューリング機能を備えていなくても、計算機における汎用のマルチスレッドオペレーティングシステムの下で数値制御装置の制御ソフトウェアを実行させることができるようになる。請求項２の制御ソフトウェア実行システムによれば、１プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムを用いたときにも、計算機と数値制御装置との間でタスクを分散して処理できるので、数値制御装置に低性能のＣＰＵを使用でき、低コストにして効率的に数値制御装置の制御ソフトウェアを実行することが出来る。請求項３の制御ソフトウェア実行システムによれば、計算機が数値制御装置への外部割り込み信号の有無を監視するので、数値制御装置のＣＰＵを無くしても割り込み処理に対処でき、数値制御装置の構成を簡略化できる。請求項４の制御ソフトウェア実行システムによれば、複数のＣＰＵカードを有するＣＰＵプールを備えるので、数値制御装置の制御ソフトウェアの処理負荷に応じてＣＰＵを獲得・解放することで処理能力を調整できる。請求項５の制御ソフトウェア実行システムによれば、ＣＰＵプールの中のＣＰＵを複数の数値制御装置のＣＰＵとして割り当てることができるので、複数の数値制御装置の制御ソフトウェアを同時に実行できる。請求項６の制御ソフトウェア実行システムによれば、ＣＰＵプールの中の１つのＣＰＵカードを割り込み検査専用のＣＰＵとして使うので、外部割り込み信号に迅速に対処することが出来る。

【図面の簡単な説明】

【図１】この発明の第１実施例の制御ソフトウェア実行システムの全体構成図である。

【図２】プロセスを生成するための流れ図である。

【図３】ホスト計算機の内部に構成されたプロセスの概念図である。

【図４】スケジューラによるスケジューリングの概念図である。

【図５】スケジューラスレッドによるスケジューリングの概念図である。

【図６】この発明の第２実施例の制御ソフトウェア実行システムの全体構成図である。

【図７】数値制御装置２００ｂ１のハードウェア構成図である。

【図８】ホスト計算機上に構築された汎用のＣＡＤ／ＣＡＭシステムの動作の概念図である。

【図９】数値制御装置とホスト計算機の間で行なわれるスレッド間通信の概念図である。

【図１０】メールボックスを介したスレッド間通信の概念図である。

【図１１】分散型共有メモリの概念図である。

【図１２】ホスト計算機と複数の数値制御装置とに分散する例の概念図である。

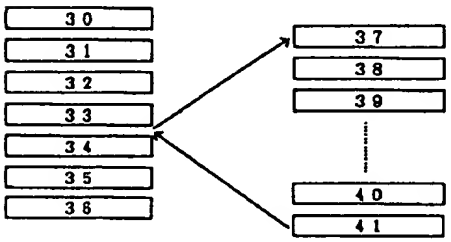
【図１３】キーボードと表示装置を備えていない数値制御装置を用いた制御ソフトウェア実行システムの全体構成図である。

【図１４】数値制御装置２００ｅ１のハードウェア構成図である。
【図１５】キーボードと表示装置を備えていない数値制御装置を用いた制御ソフトウェア実行システムの動作を示す概念図である。
【図１６】ビットマップディスプレイを用いた制御ソフトウェア実行システムの全体構成図である。
【図１７】ビットマップディスプレイを用いた制御ソフトウェア実行システムにおける表示処理の概念図である。
【図１８】マウスを備えた制御ソフトウェア実行システムの全体構成図である。
【図１９】マウスを備えた制御ソフトウェア実行システムにおける表示処理の概念図である。
【図２０】この発明の第３実施例の数値制御装置２００ｉ１のハードウェア構成図である。
【図２１】この発明の第４実施例における制御ソフトウェア実行システムの全体構成図である。
【図２２】ＣＰＵプールの内部構成図である。
【図２３】新たなＣＰＵカードを獲得するときの動作の概念図である。
【図２４】この発明の変形実施例の制御ソフトウェア実行システムの全体構成図である。
【図２５】数値制御装置２００ｋ１のハードウェア構成図である。
【図２６】数値制御装置２００ｑ１のハードウェア構成図である。
【図２７】従来の数値制御装置のハードウェア構成図である。
【図２８】従来の数値制御装置の処理手順を示す説明図である。
【図２９】割り込み処理の概念図である。
【図３０】各タスク動作の時間的関係を示すタイムチャートである。
【図３１】ワークステーションを利用した制御ソフトウェア開発システムの構成図である。
【図３２】制御ソフトウェアの開発手順を示す流れ図である。
【図３３】クライアントワークステーションのメモリ態様図である。
【図３４】各プロセスが同じ論理アドレス空間を利用できることを示す原理図である。
【図３５】１つのプロセスの論理アドレス空間内の領域図である。
【図３６】タイムシェアリングオペレーティングシステムにおけるプロセスモデルの例示図である。
【図３７】マルチスレッドオペレーティングシステムにおけるモデルの例示図である。
【図３８】１つのプロセスの論理アドレス空間内の別の領域図である。

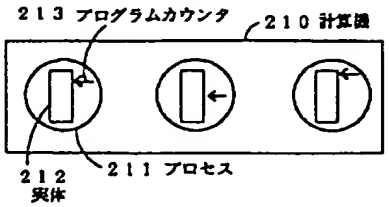
【符号の説明】

１　メインＣＰＵ
２　システムバス
３　ＲＯＭ
４　ＲＡＭ
５　ＳＩＯインタフェース部
６　表示装置
７　キーボード
８　サーボ制御部
９　サーボアンプ
１０　サーボモータ
１１　プログラムコントローラ部
１２　外部割り込み信号
１３２　アドレス変換機構
１１３３　フラッシュＲＯＭ
Ｓ１～Ｓ１４　制御ソフトウェア実行システム
２００ａ１，…，２００ｑ１　数値制御装置
２００ｋｎ，…，２００ｑｎ　数値制御装置
２０１　ホスト計算機
２０２　ハードディスク
２０３，２０３ａ　表示装置
２０４　キーボード
２０７　ネットワークインタフェース
２０５，２０８　内部ネットワーク
２０６　外部ネットワーク
２０７　ネットワークインタフェース
３０１　ＣＰＵプール
３０２　ＣＰＵカード
２０７０　ゲートウェイ部
Ｍ　　マウス

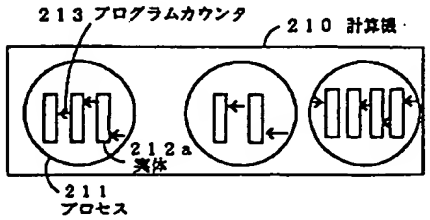
【図２９】



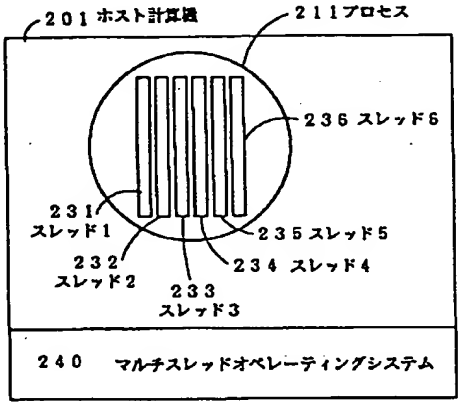
【図３６】



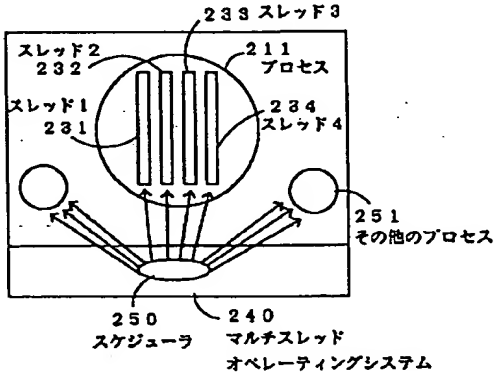
【図３７】



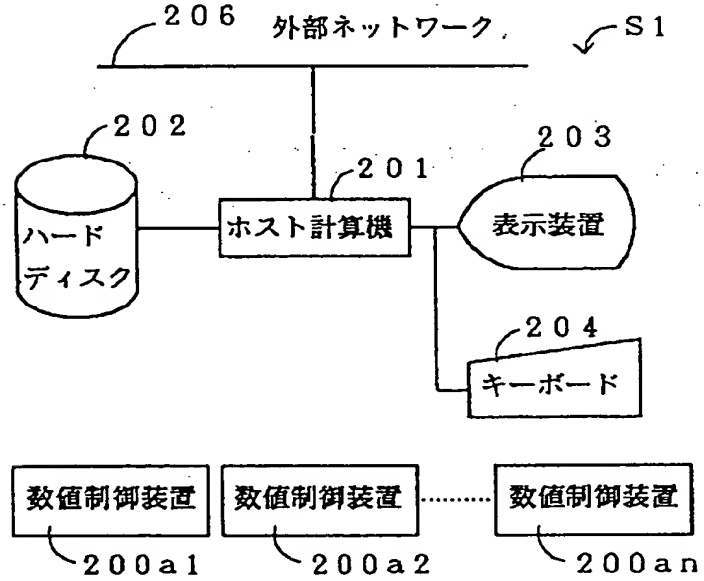
【図3】


201 ホスト計算機
211プロセス
236 スレッド6
231
スレッド1
235 スレッド5
232
スレッド2
234 スレッド4
233
スレッド3
240 マルチスレッドオペレーティングシステム

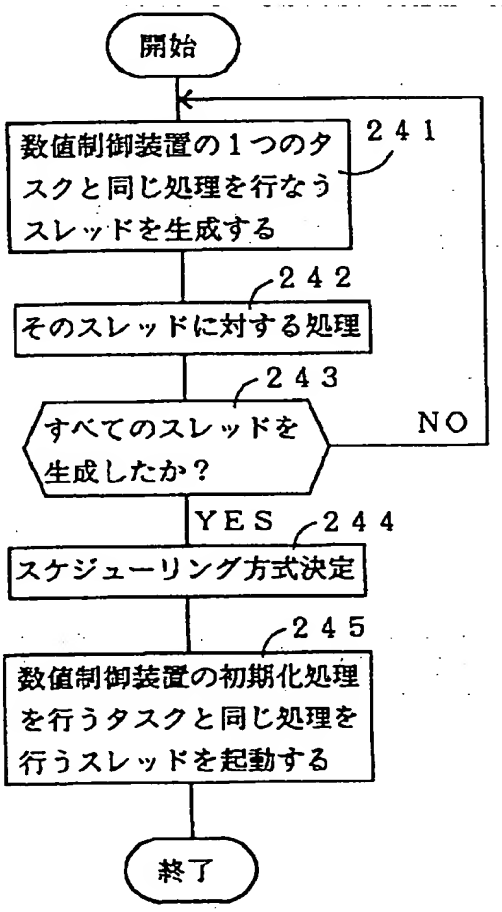
【図4】


233 スレッド3
スレッド2
232
211
プロセス
スレッド1
231
234
スレッド4
251
その他のプロセス
250
スケジューラ
240
マルチスレッド
オペレーティングシステム

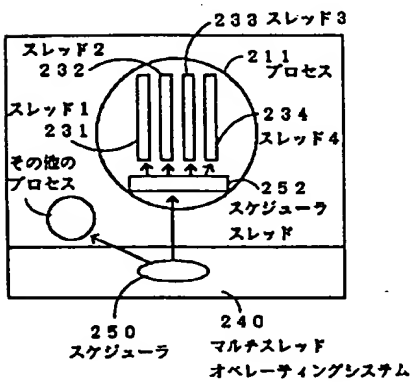
【図1】


206 外部ネットワーク
S1
202
ハード
ディスク
201
ホスト計算機
203
表示装置
204
キーボード
数値制御装置
200a1
数値制御装置
200a2
数値制御装置
200an

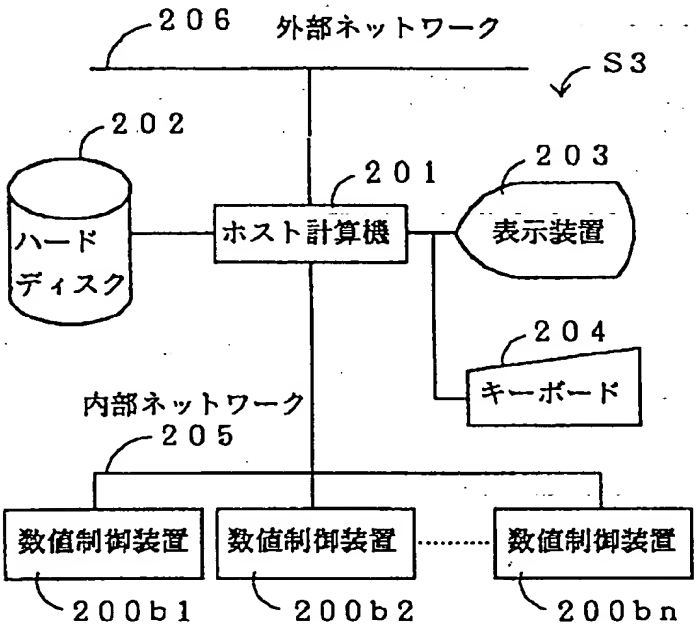
【図2】


開始
数値制御装置の1つのタスクと同じ処理を行なうスレッドを生成する
241
そのスレッドに対する処理
242
すべてのスレッドを生成したか？
243
NO
YES
スケジューリング方式決定
244
数値制御装置の初期化処理を行うタスクと同じ処理を行うスレッドを起動する
245
終了

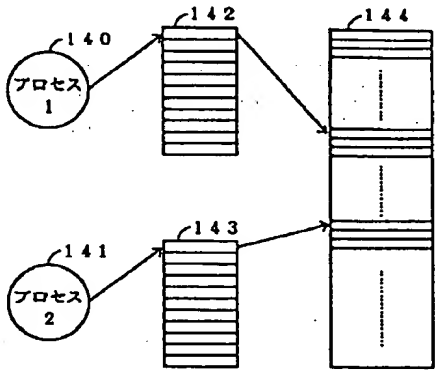
【図5】


233 スレッド3
スレッド2
232
211
プロセス
スレッド1
231
234
スレッド4
その他の
プロセス
252
スケジューラ
スレッド
250
スケジューラ
240
マルチスレッド
オペレーティングシステム

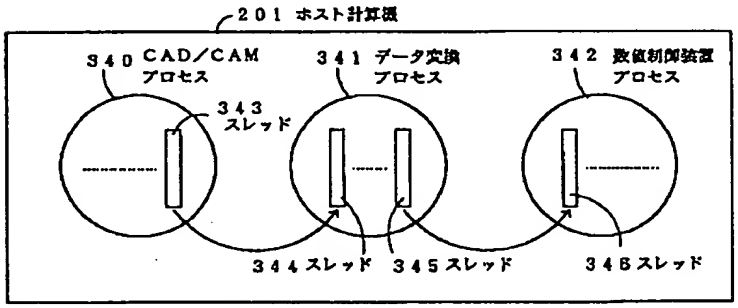
【図６】


２０６ 外部ネットワーク
S３
２０２
２０１
２０３
ハード
ディスク
ホスト計算機
表示装置
２０４
キーボード
内部ネットワーク
２０５
数値制御装置
数値制御装置
数値制御装置
２００ｂ１
２００ｂ２
２００ｂｎ

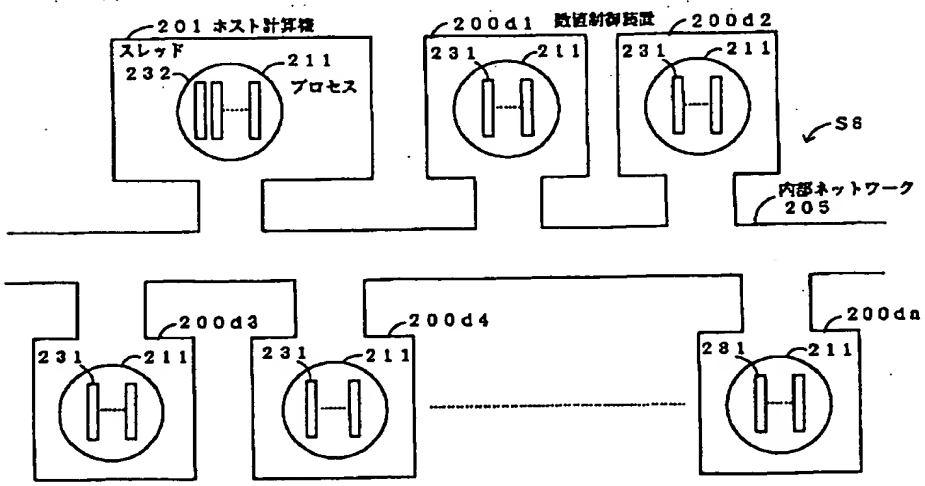
【図３４】


１４０
プロセス
１
１４２
１４４
１４１
プロセス
２
１４３

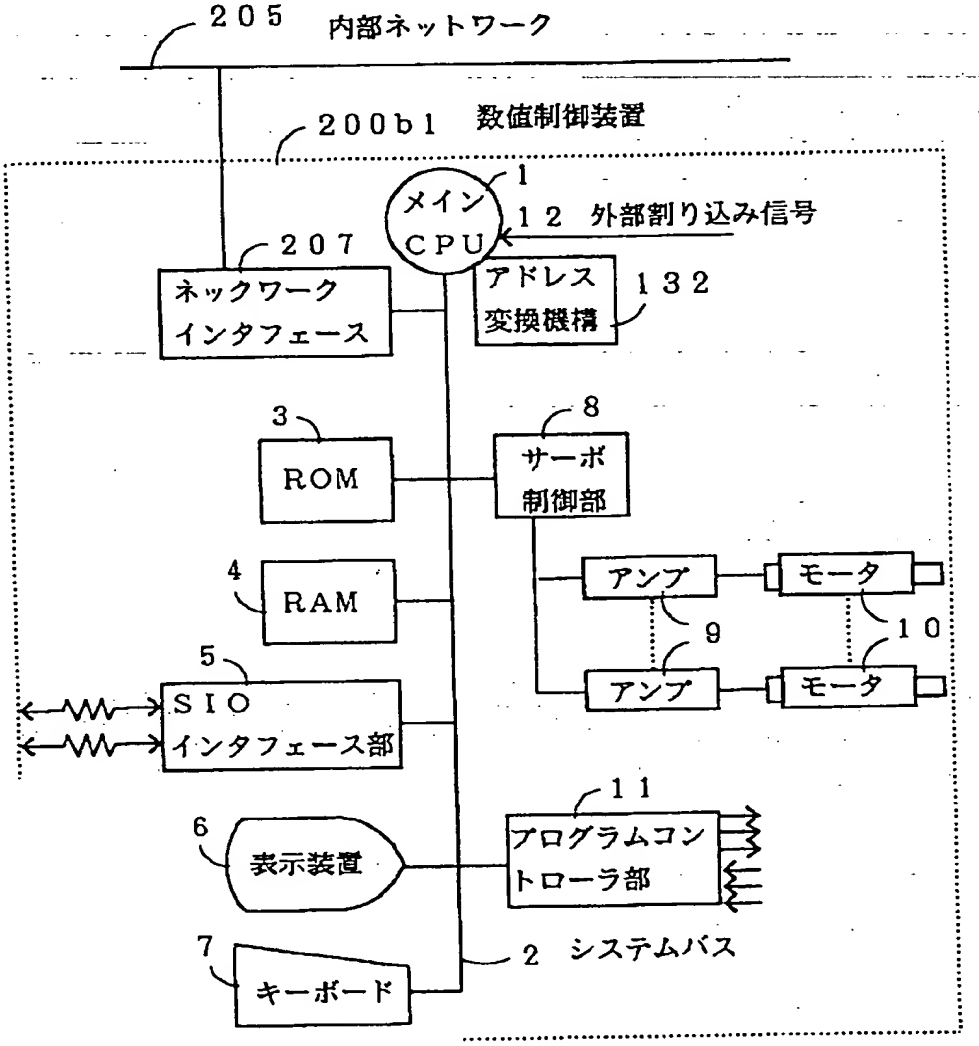
【図８】


２０１ ホスト計算機
３４０ CAD/CAM
プロセス
３４１ データ変換
プロセス
３４２ 数値制御装置
プロセス
３４３
スレッド
３４４スレッド
３４５スレッド
３４６スレッド

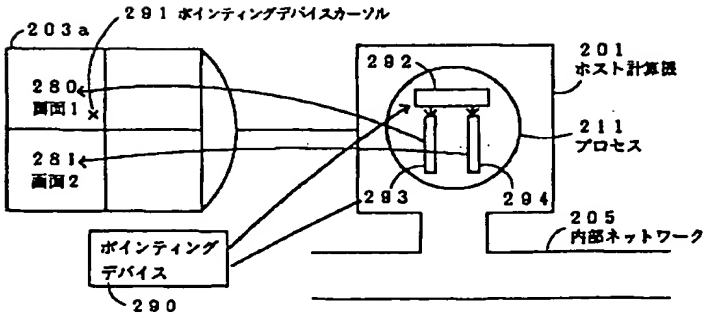
【図１２】


２０１ ホスト計算機
スレッド
２３２
２１１
プロセス
２００ｄ１ 数値制御装置
２００ｄ２
２３１
２１１
S８
内部ネットワーク
２０５
２００ｄ３
２００ｄ４
２００ｄｎ

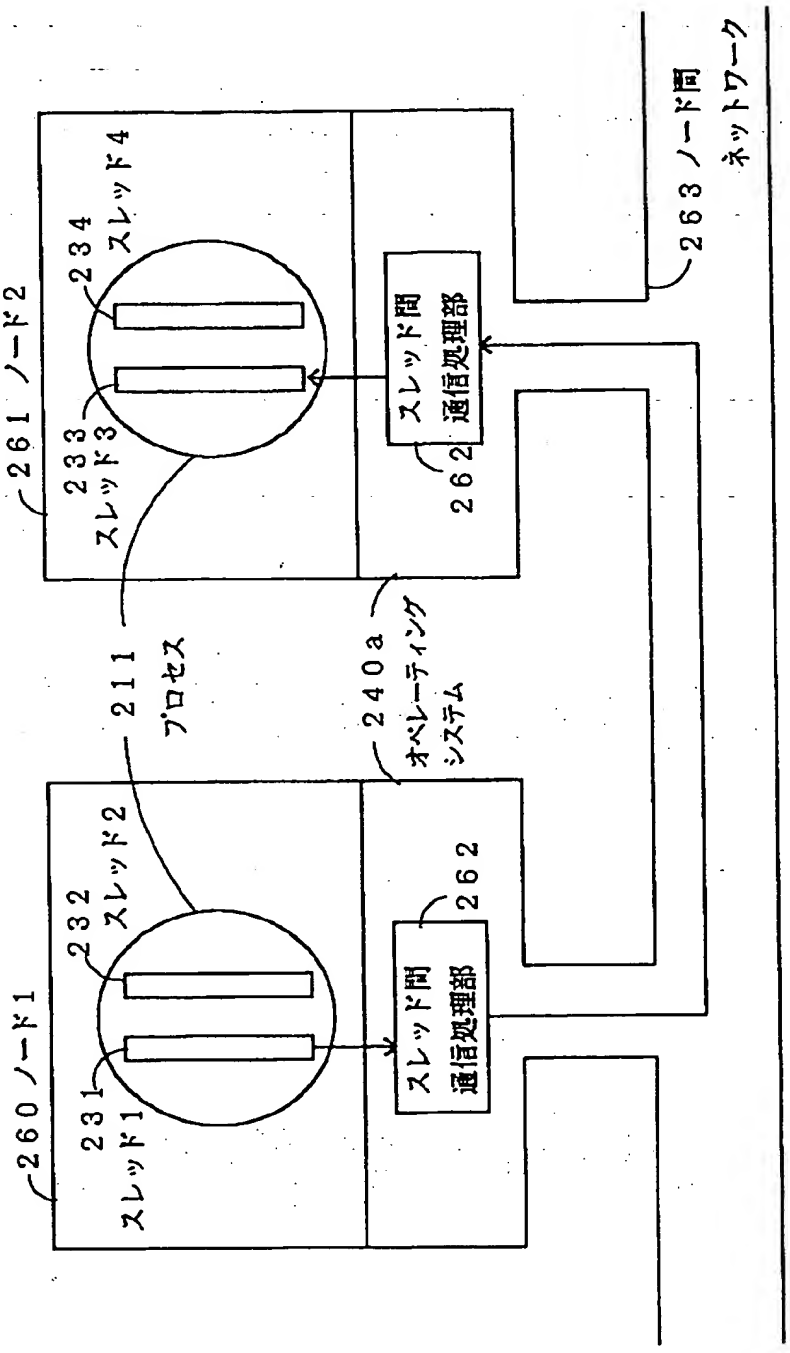
【図7】


205 内部ネットワーク
200b1 数値制御装置
1
メイン CPU
12 外部割り込み信号
207
ネックワーク インタフェース
アドレス 変換機構
132
3
ROM
8
サーボ 制御部
4
RAM
アンプ
モータ
9
10
5
SIO インタフェース部
アンプ
モータ
11
6
表示装置
プログラムコントローラ部
7
キーボード
2 システムバス

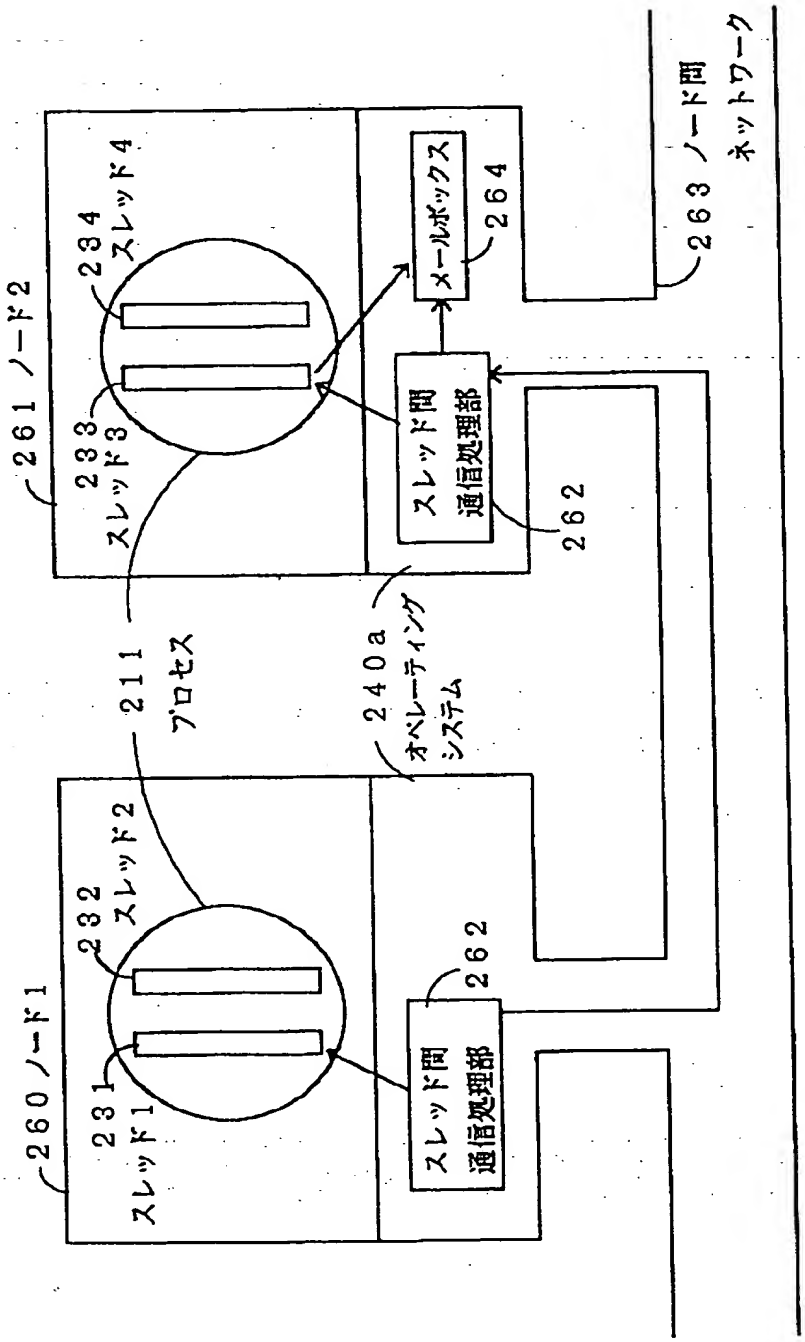
【図19】


203a
291 ポインティングデバイスカーソル
280 画面1
281 画面2
292
201 ホスト計算機
211 プロセス
293
294
205 内部ネットワーク
ポインティング デバイス
290

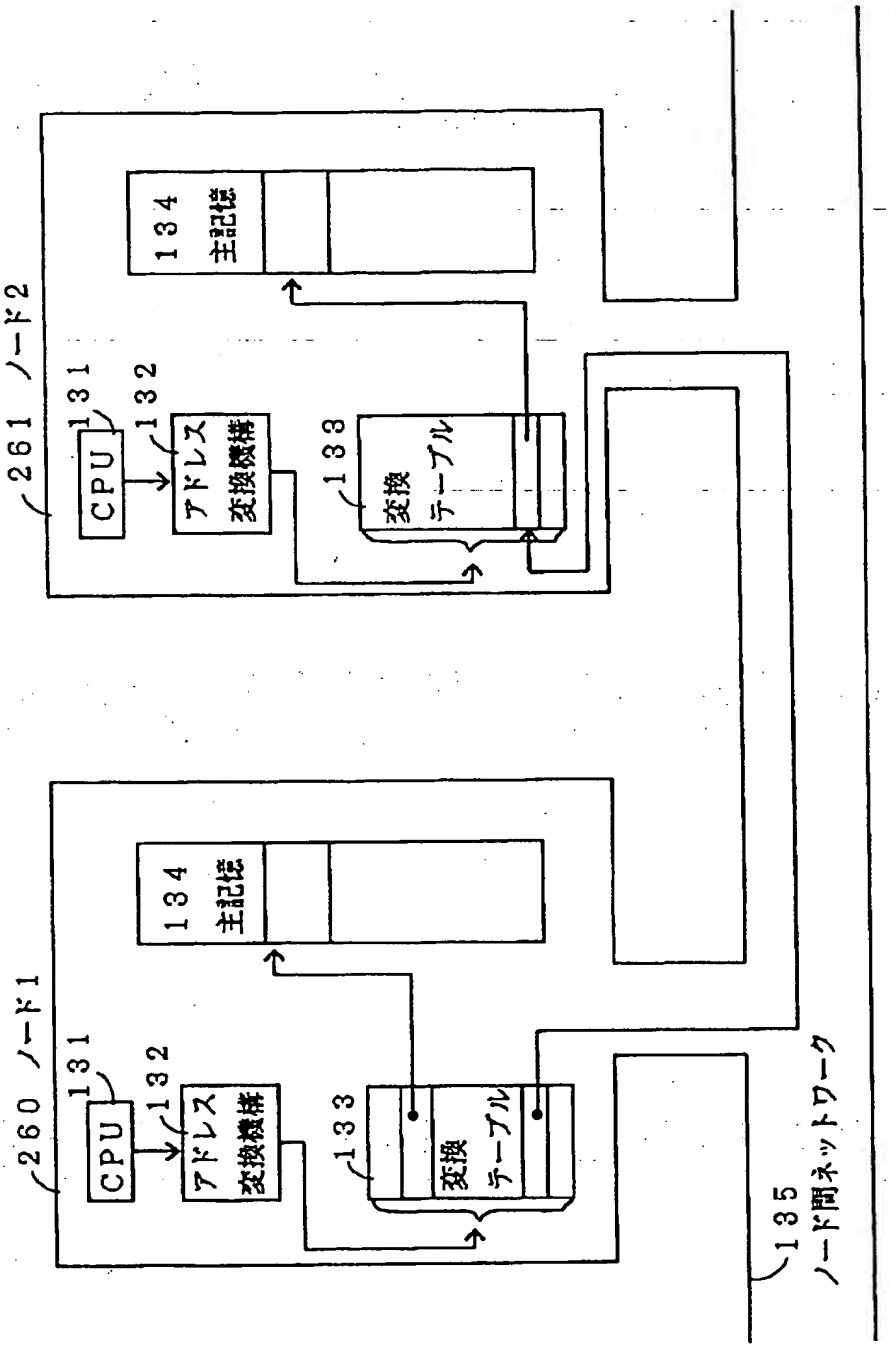
【図9】


260 ノード1
231 スレッド1
232 スレッド2
211 プロセス
261 ノード2
233 スレッド3
234 スレッド4
スレッド間 通信処理部
262
240a オペレーティング システム
263 ノード間 ネットワーク

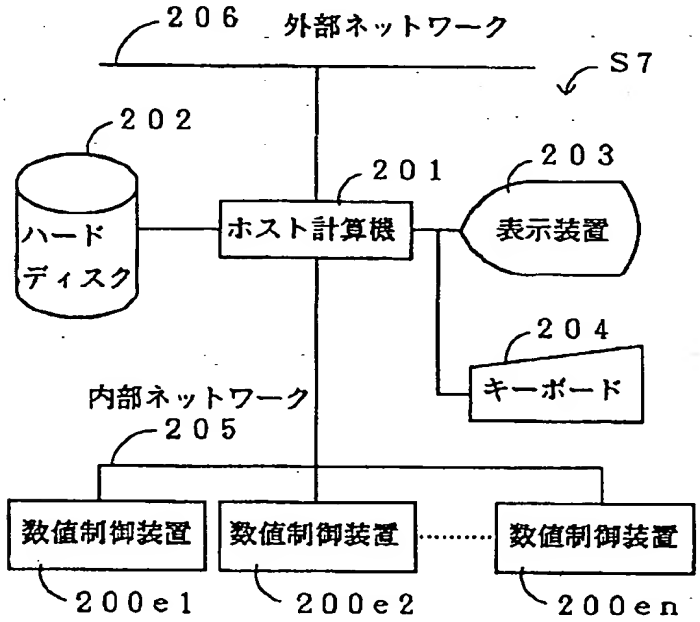
【図10】


260 ノード1
231 スレッド1
232 スレッド2
211 プロセス
261 ノード2
233 スレッド3
234 スレッド4
スレッド間通信処理部
262
スレッド間通信処理部
262
240a オペレーティングシステム
メールボックス
264
263 ノード間ネットワーク

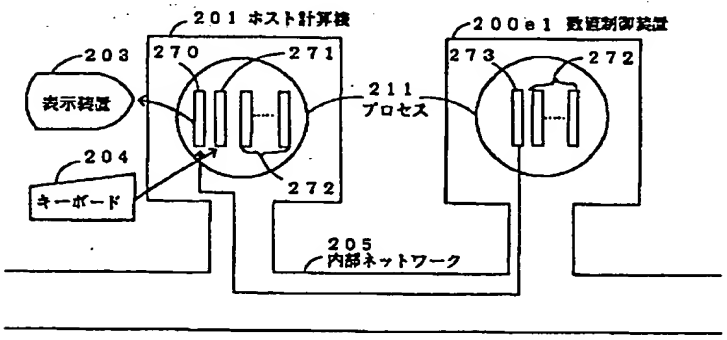
【図１１】


260 ノード1
261 ノード2
131
CPU
132
アドレス変換機構
133
変換テーブル
134
主記憶
135
ノード間ネットワーク

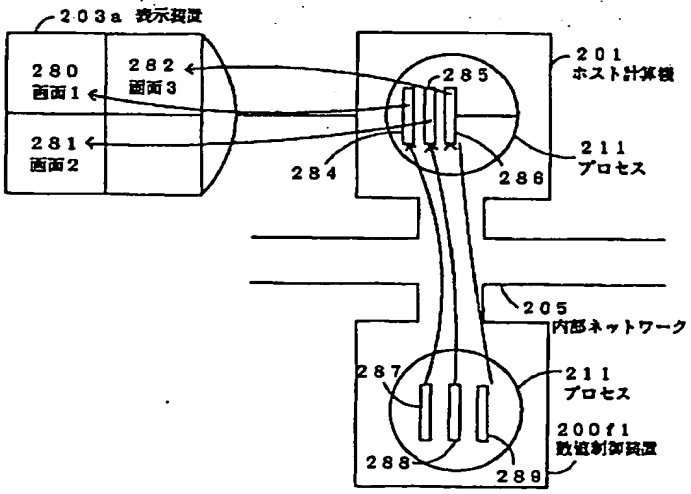
【図１３】



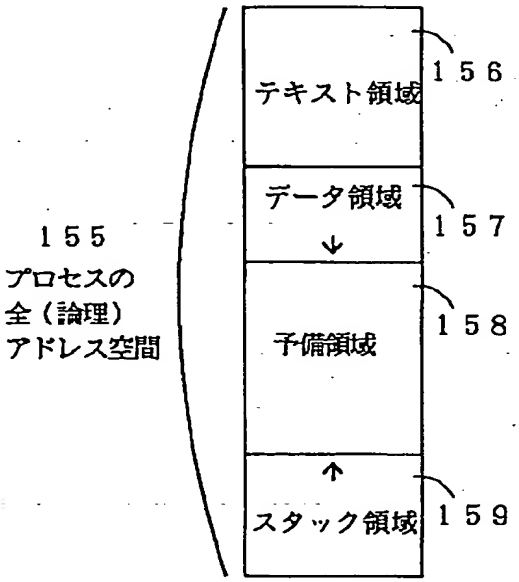
【図１５】



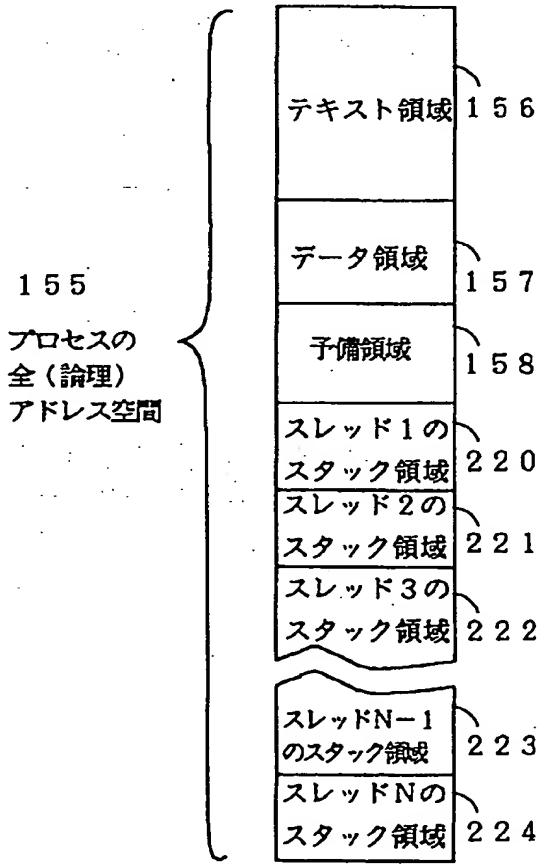
【図１７】



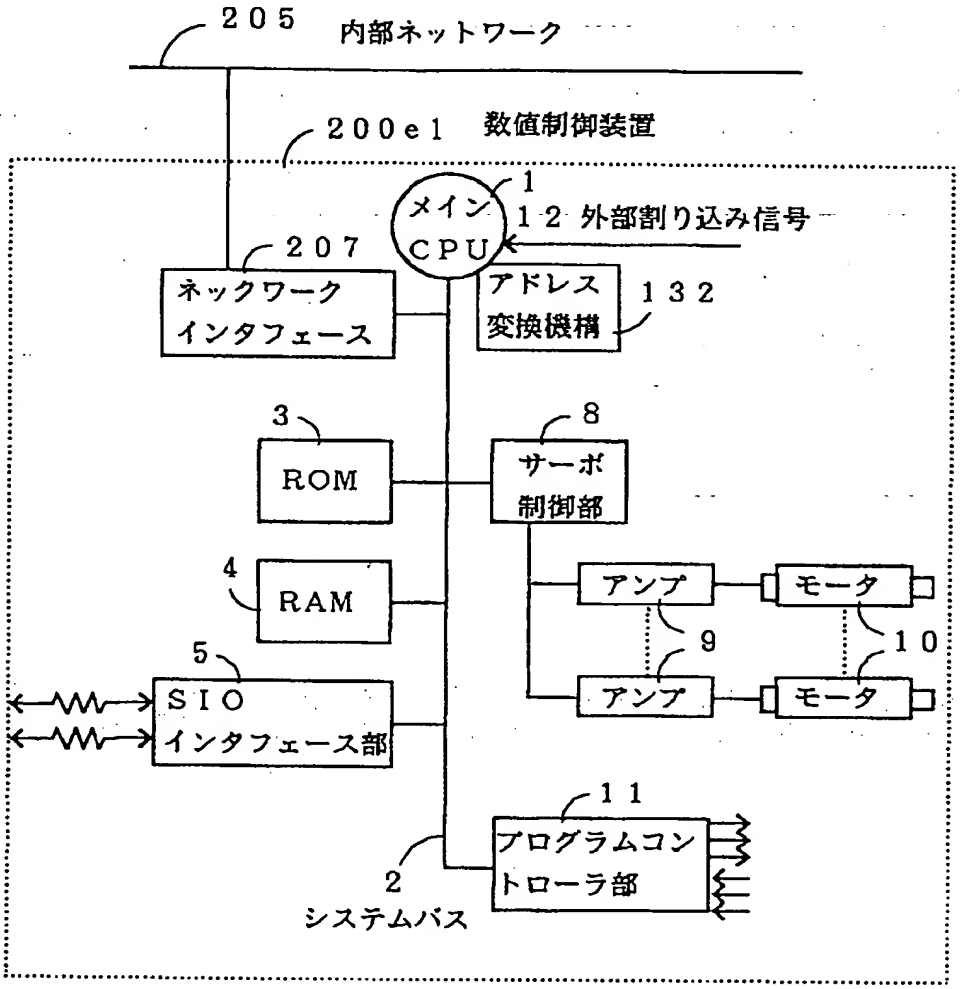
【図３５】



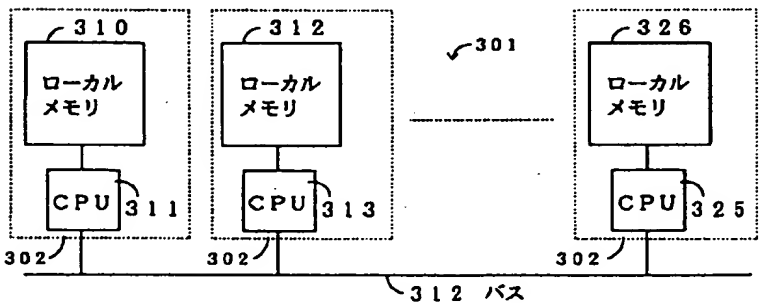
【図３８】



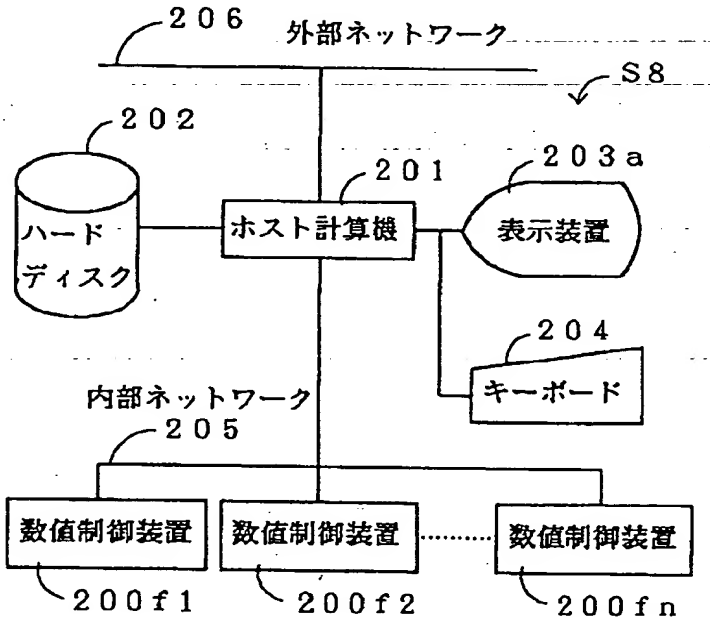
【図１４】


２０５
内部ネットワーク
２００ｅ１
数値制御装置
１
メイン
ＣＰＵ
１２ 外部割り込み信号
２０７
ネックワーク
インタフェース
アドレス
変換機構
１３２
３
ＲＯＭ
８
サーボ
制御部
４
ＲＡＭ
アンプ
モータ
９
１０
５
ＳＩＯ
インタフェース部
アンプ
モータ
１１
プログラムコン
トローラ部
２
システムバス

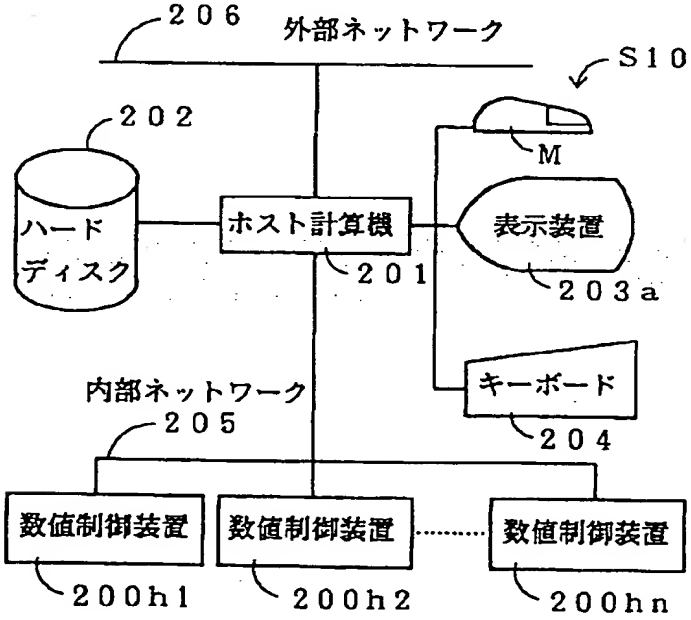
【図２２】


３１０
ローカル
メモリ
ＣＰＵ
３１１
３１２
ローカル
メモリ
ＣＰＵ
３１３
３０１
３２６
ローカル
メモリ
ＣＰＵ
３２５
３０２
３０２
３０２
３１２ バス

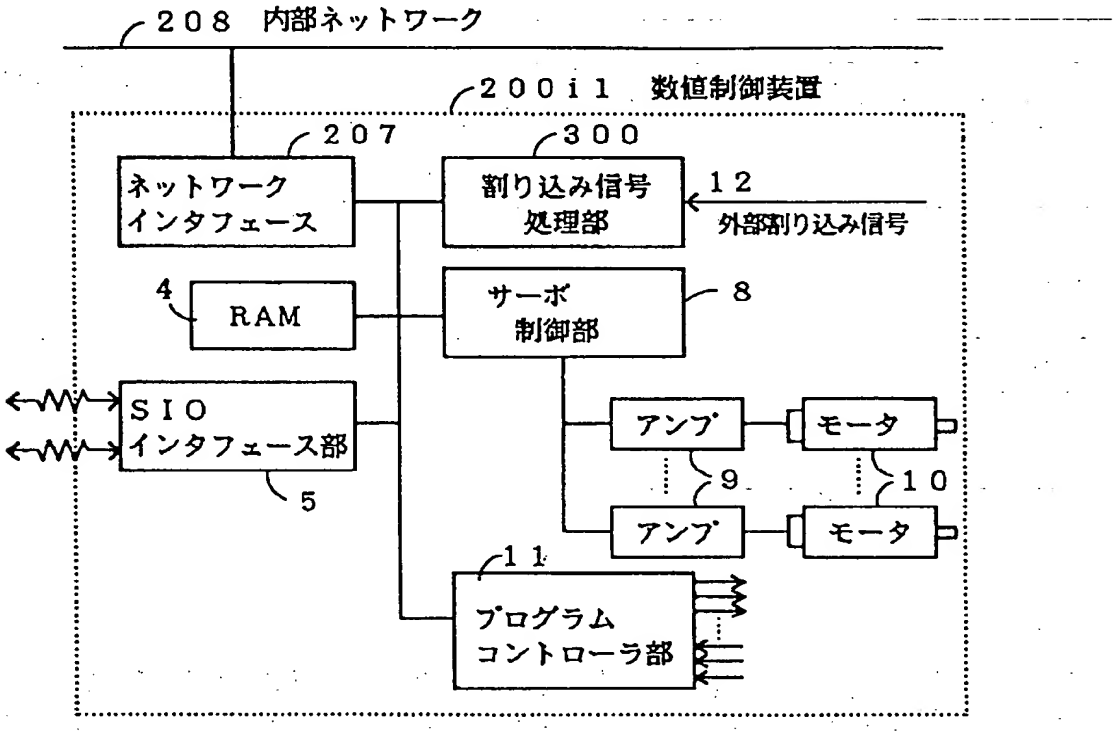
【図１６】


２０６
外部ネットワーク
Ｓ８
２０２
ハード
ディスク
２０１
ホスト計算機
２０３ａ
表示装置
２０４
キーボード
内部ネットワーク
２０５
数値制御装置
２００ｆ１
数値制御装置
２００ｆ２
数値制御装置
２００ｆｎ

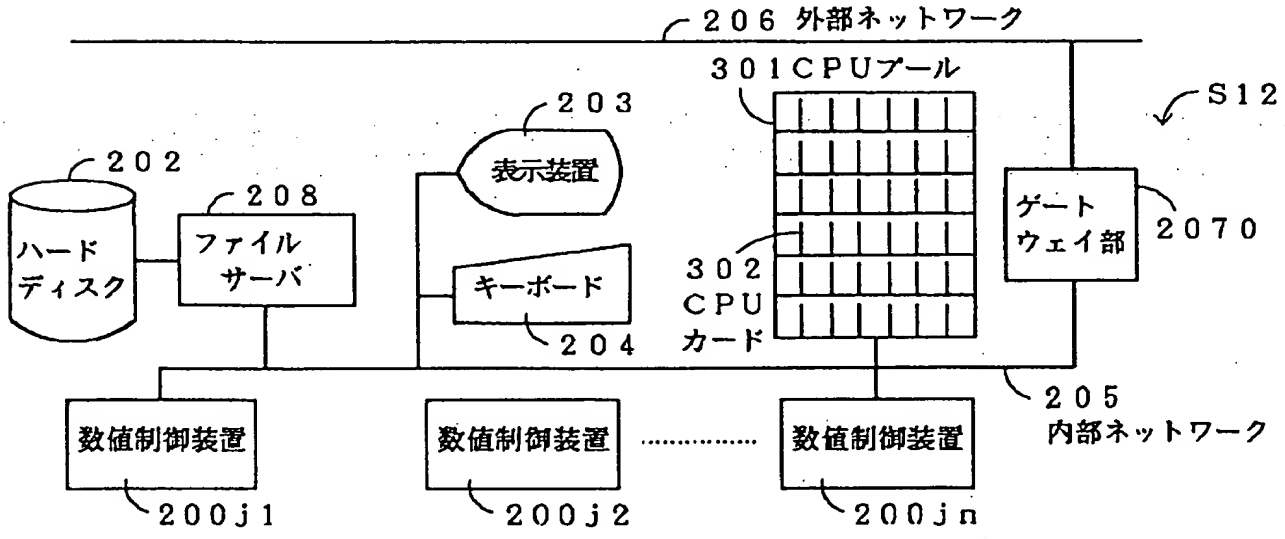
【図１８】


２０６
外部ネットワーク
Ｓ１０
２０２
ハード
ディスク
Ｍ
ホスト計算機
２０１
表示装置
２０３ａ
キーボード
２０４
内部ネットワーク
２０５
数値制御装置
２００ｈ１
数値制御装置
２００ｈ２
数値制御装置
２００ｈｎ

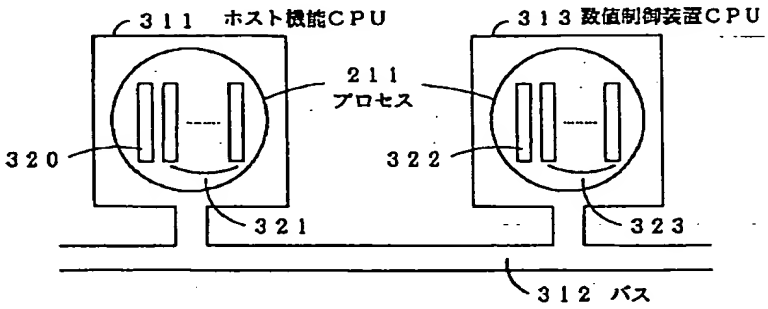
【図２０】


２０８　内部ネットワーク
２００ｉ１　数値制御装置
２０７
３００
ネットワーク
インタフェース
割り込み信号
処理部
１２
外部割り込み信号
４
RAM
サーボ
制御部
８
ＳＩＯ
インタフェース部
５
アンプ
モータ
９
１０
アンプ
モータ
１１
プログラム
コントローラ部

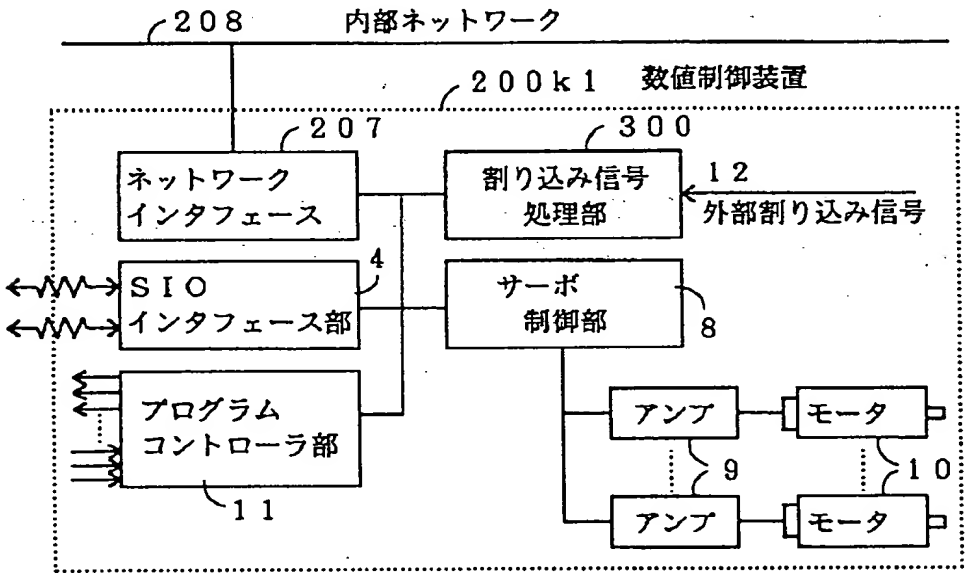
【図２１】


２０６ 外部ネットワーク
３０１ＣＰＵプール
２０３
Ｓ１２
２０２
２０８
表示装置
ゲート
ウェイ部
２０７０
ハード
ディスク
ファイル
サーバ
３０２
ＣＰＵ
カード
キーボード
２０４
２０５
内部ネットワーク
数値制御装置
数値制御装置
数値制御装置
２００ｊ１
２００ｊ２
２００ｊｎ

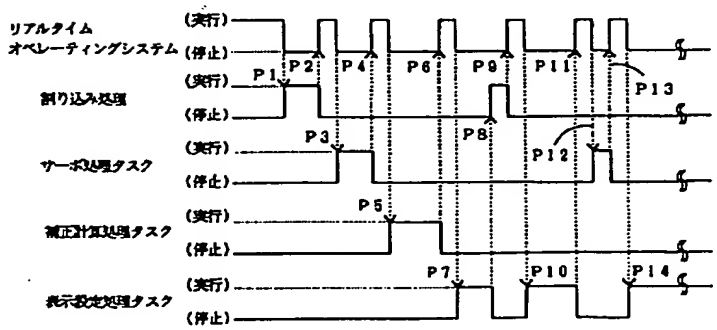
【図２３】

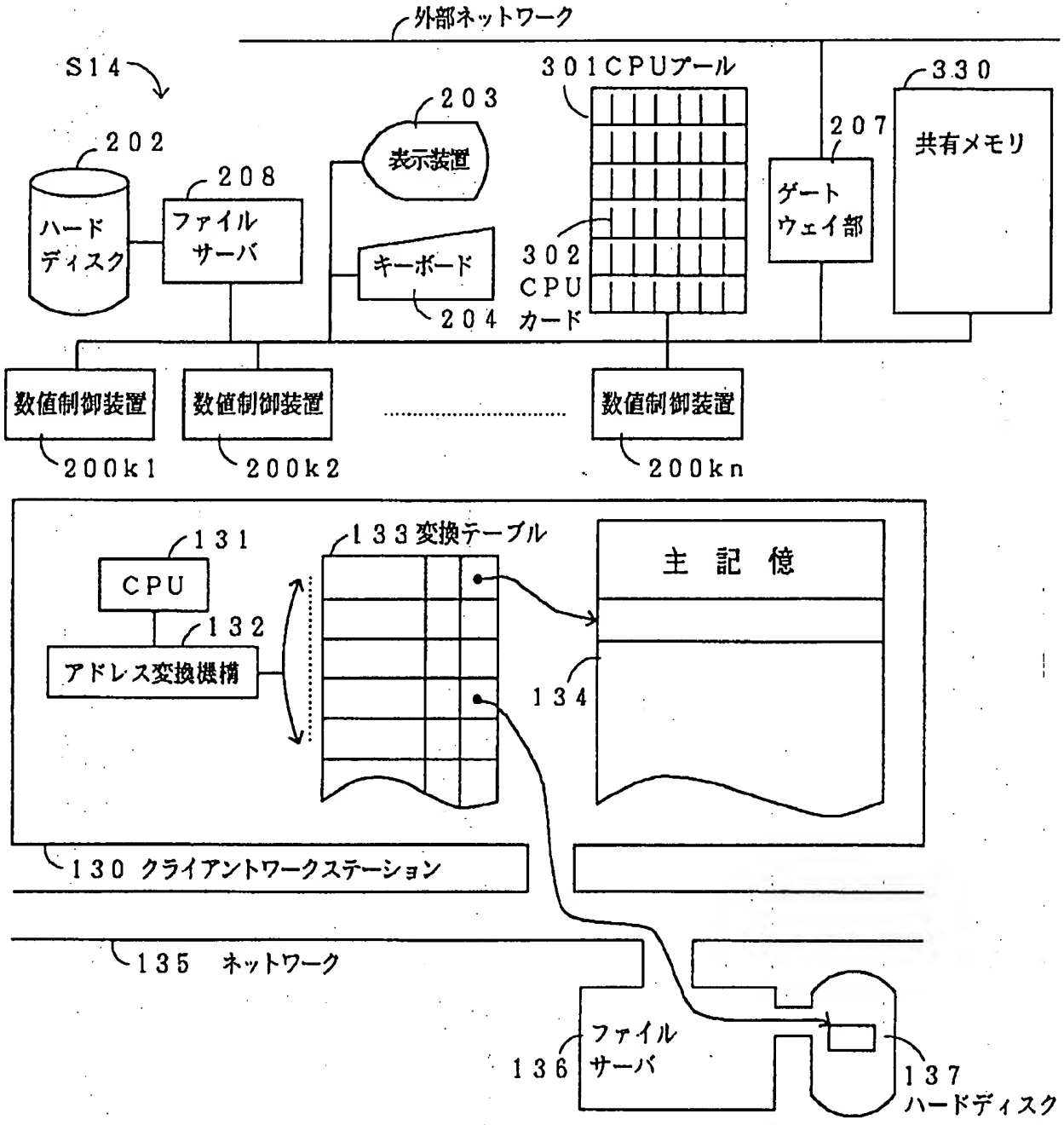

３１１　ホスト機能ＣＰＵ
３１３ 数値制御装置ＣＰＵ
２１１
プロセス
３２０
３２２
３２１
３２３
３１２ バス

【図２５】


２０８　内部ネットワーク
２００ｋ１　数値制御装置
２０７
３００
ネットワーク
インタフェース
割り込み信号
処理部
１２
外部割り込み信号
４
ＳＩＯ
インタフェース部
サーボ
制御部
８
プログラム
コントローラ部
１１
アンプ
モータ
９
１０
アンプ
モータ

【図３０】

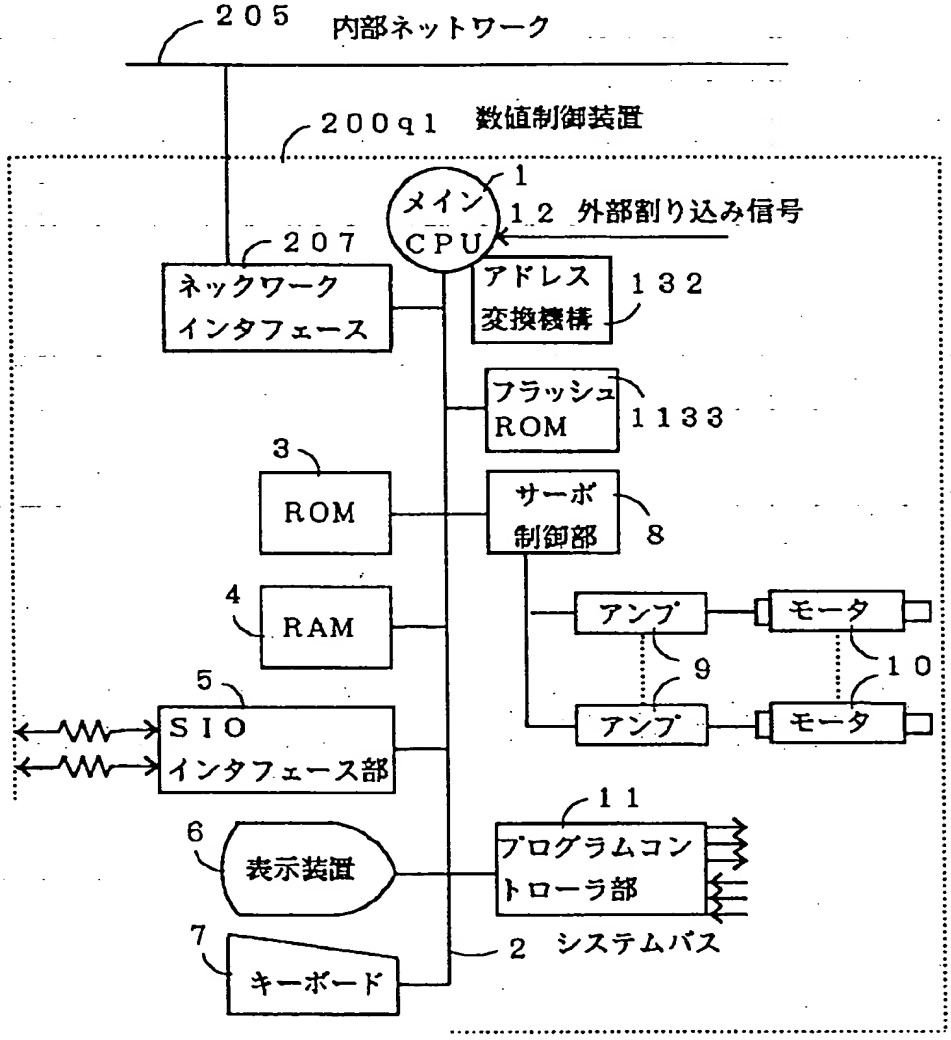

リアルタイム
オペレーティングシステム
(実行)
(停止)
P1
P2
P4
P6
P9
P11
P13
割り込み処理
P3
P8
P12
サーボ処理タスク
P5
補正計算処理タスク
P7
P10
P14
表示設定処理タスク


外部ネットワーク
S14
301CPUプール
330
203
207
共有メモリ
202
208
表示装置
ハード
ディスク
ファイル
サーバ
ゲート
ウェイ部
302
CPU
カード
キーボード
204
数値制御装置
数値制御装置
数値制御装置
200k1
200k2
200kn
131
133変換テーブル
主 記 憶
CPU
132
アドレス変換機構
134
130 クライアントワークステーション
135 ネットワーク
ファイル
136 サーバ
137
ハードディスク

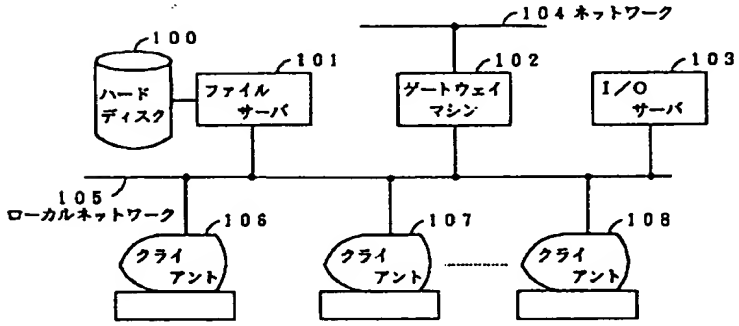
【図24】

【図33】

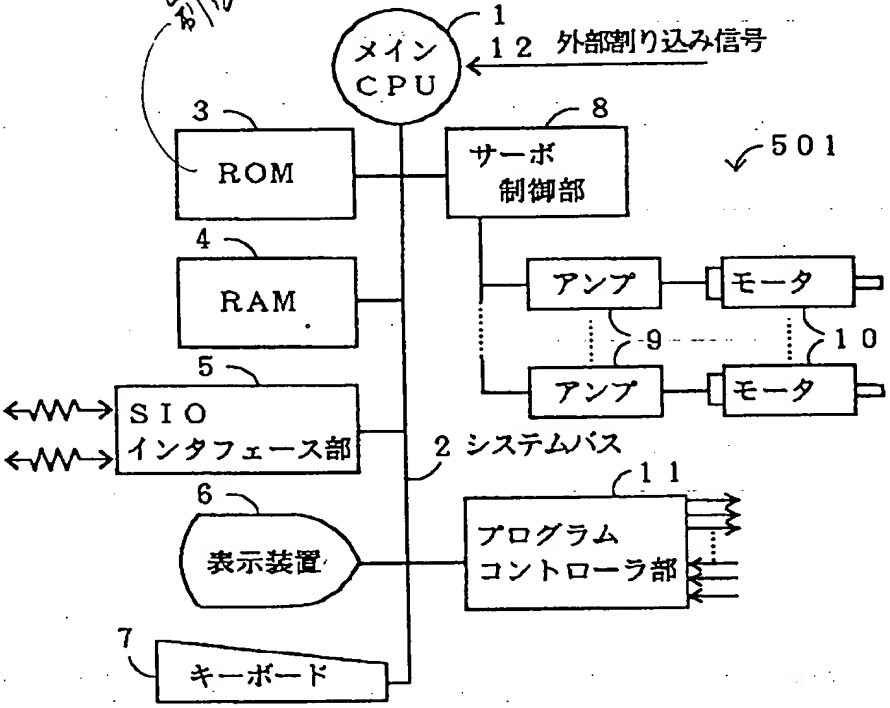
【図２６】


２０５ 内部ネットワーク
２００ｑ１ 数値制御装置
１
メイン CPU
１２ 外部割り込み信号
２０７
ネックワーク インタフェース
アドレス 変換機構
１３２
フラッシュ ROM
１１３３
３
ROM
サーボ 制御部
８
４
RAM
アンプ
モータ
９
１０
５
ＳＩＯ インタフェース部
アンプ
モータ
１１
６
表示装置
プログラムコントローラ部
７
キーボード
２ システムバス

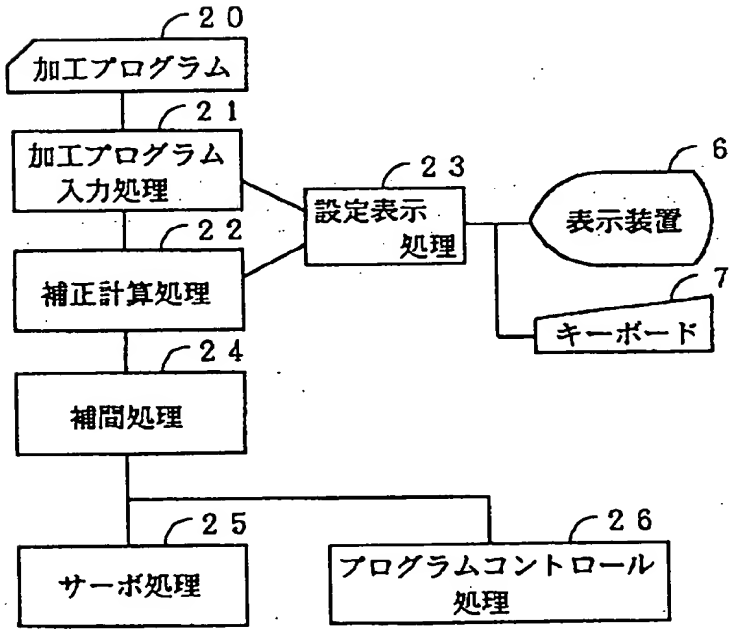
【図３１】


１０４ ネットワーク
１００
ハードディスク
１０１
ファイルサーバ
１０２
ゲートウェイマシン
１０３
I/O サーバ
１０５
ローカルネットワーク
１０６
クライアント
１０７
クライアント
１０８
クライアント

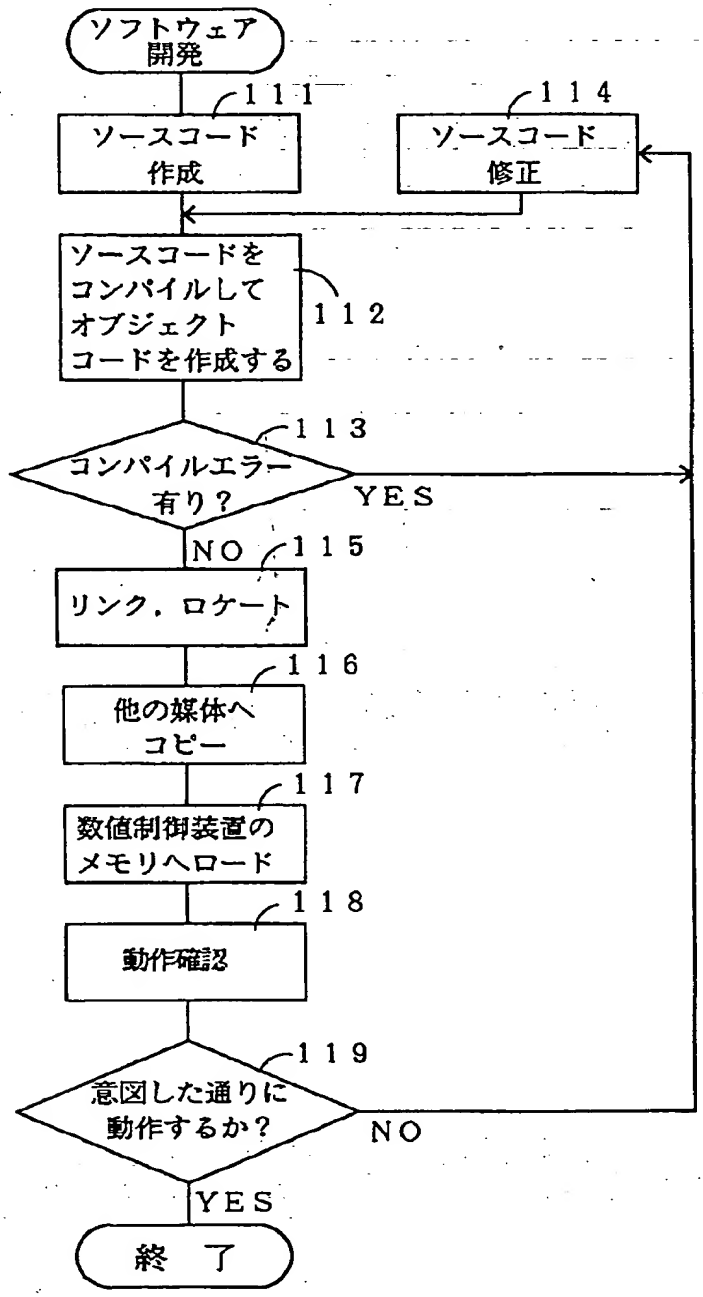
【図２７】


メイン
CPU
1
12 外部割り込み信号
3
ROM
8
サーボ
制御部
501
4
RAM
アンプ
モータ
9
10
アンプ
モータ
5
SIO
インタフェース部
2 システムバス
11
6
表示装置
プログラム
コントローラ部
7
キーボード

【図２８】


20
加工プログラム
21
加工プログラム
入力処理
23
設定表示
処理
6
表示装置
22
補正計算処理
7
キーボード
24
補間処理
25
サーボ処理
26
プログラムコントロール
処理

【図３２】


ソフトウェア開発
１１１
ソースコード作成
１１４
ソースコード修正
ソースコードをコンパイルしてオブジェクトコードを作成する
１１２
１１３
コンパイルエラー有り？
YES
NO
１１５
リンク，ロケート
１１６
他の媒体へコピー
１１７
数値制御装置のメモリへロード
１１８
動作確認
１１９
意図した通りに動作するか？
NO
YES
終　了